SPY: Enhancing Privacy with Synthetic PII Detection Dataset

Maksim Savkin[1,†] **Timur Ionov**[2,3,†] **Vasily Konovalov**[4,1]
[1]Moscow Institute of Physics and Technology
[2]MTS AI [3]ITMO University [4]AIRI
{savkin.mk, vasily.konovalov}@phystech.edu t.ionov@mts.ai

Abstract

We introduce the SPY dataset: a novel synthetic dataset designed for Personal Identifiable Information (PII) detection, underscoring the importance of safeguarding PII in modern data processing. Our approach innovates by using large language models (LLMs) to generate a dataset that emulates real-world PII scenarios. We evaluate the dataset's quality and position it as a reliable benchmark for PII detection.. Comparative analyses reveal that while PII detection 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[1], thereby enabling further research and development in this field.

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 government 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, 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 described 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)	Lucy Cechtelar lives at 426 Jordy Lodge Cartwrightshire, SC 88120-6700.

Table 1: Examples of **a)** NER entities; **b)** PII entities. All examples of personal information provided are generated using the Faker library (Faraglia, 2014).

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) use a dataset that contains real PII entities and substitute them with fake ones; (2) devise a methodology to generate a completely PII-focused dataset from scratch and then replace the placeholders of PII with fake entities generated by a tool such as Faker (Faraglia, 2014), see Section 3 for more details. The benefit of the former approach is that it maintains the data's characteristics. The drawback is in ensuring that all genuine PII entities have been accurately replaced.

In our work, we opt for the second approach. We used Faker to create artificial PII entities and Llama-3-70B (AI@Meta, 2024) 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

† These authors contributed equally to this work.

[1] https://github.com/LogicZMaksimka/SPY_Dataset

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 236–246
April 30 - May 1, 2025 ©2025 Association for Computational Linguistics

their desired distribution or balance.

Our contributions can be summarized as follows.

- We present a methodology for developing the 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 the SPY dataset containing 4,491 medical consultations and 4,197 questions in the legal domain, which 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

Knowledge-based approaches for safeguarding PII like `regexp` achieve fair accuracy in identifying PII that have a strict and template-based format, but 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 in various contexts, particularly when dealing with unstructured data such as 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. 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, **NER-PII** (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 by manually annotation of The stack (Kocetkov et al., 2023) dataset. Specifically, it targets the identification of PII in programming contexts, making it less suitable for broader text-based PII scenarios.

The **AI4Privacy**[3] is a synthetic PII dataset created using proprietary algorithms. It spans six languages and eight jurisdictions, with 63 PII classes, making it one of the most comprehensive datasets available. However, its proprietary nature limits transparency, making 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 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 essays are written in response to a single assignment prompt, which limits the diversity of the data. Second, only 30% of the dataset is publicly available for training, with the remaining 70% reserved for testing, making it unsuitable for a comprehensive evaluation (see Figure 2 for detailed statistics). Although 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.

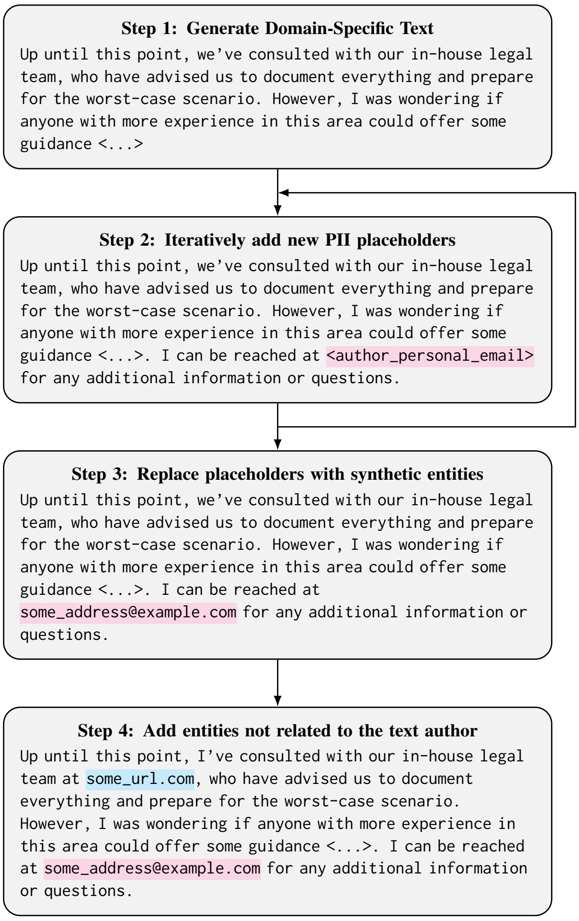
2.1 Synthetic NER Generation

Although research on PII datasets is limited due to privacy concerns, significant work has been done on generating synthetic NER datasets that share a similar format with PII data.

A notable approach is described 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 are 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.

[2] `https://hf.co/datasets/bigcode/bigcode-pii-dataset`

[3] `https://hf.co/datasets/ai4privacy/pii-masking-300k`



Step 1: Generate Domain-Specific Text

Up until this point, we've consulted with our in-house legal team, 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 <...>

Step 2: Iteratively add new PII placeholders

Up until this point, we've consulted with our in-house legal team, 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 <...>. I can be reached at <author_personal_email> for any additional information or questions.

Step 3: Replace placeholders with synthetic entities

Up until this point, we've consulted with our in-house legal team, 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 <...>. I can be reached at some_address@example.com for any additional information or questions.

Step 4: Add entities not related to the text author

Up until this point, I've consulted with our in-house legal team at some_url.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 <...>. I can be reached at some_address@example.com for any additional information or questions.

Figure 1: Multi-step prompting procedure. Red selection – author's personal data (PII); blue selection – NER entities not directly related to the text author. Prompts used in Steps 1–4 are shown in Figures 6,8,9 and 10, respectively.

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., 2024) leverage GPT-3.5 to annotate large text corpora, such as The Pile (Gao et al., 2021) 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

Although direct prompting of LLMs to annotate text data has proven effective for datasets rich in NER entities (Zhou et al., 2024; Bogdanov et al., 2024; Zaratiana et al., 2024), 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 the 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 (Langdon et al., 2024), only 24% of all essays contain any personal data, and six of the seven entity types have fewer than 110 samples (see Figure 2), leading to class imbalance and limited representation. To address those constraints, we generate texts that contain placeholders for predetermined sets of personal entities. Then we replace these placeholders with PII entities generated by Faker - an open-source python library that generates realistic synthetic entities. It can produce a wide range of data types, including names, addresses, emails, dates, and more, supporting multiple locales and customization.

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 is 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 contrast, all non-PII entities are unrelated to the author. These limitations 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

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

the prompt in Figure 6. Back when we conducted the main experiments, Llama-3-70B was one of the best 70B models available for instruction following. It performed well across the required data manipulations from the prompt, handling the diverse task requirements effectively. We did not opt for proprietary models due to budget constraints, which influenced our decision to use Llama-3-70B for this project. We did not opt for proprietary models due to budget constraints.

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. The personalities were generated using the prompt shown in Figure 7.

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 8. Although iterative text updates can be performed using CoD prompts (Adams et al., 2023), we found that Llama-3-70B struggled to apply multiple updates in a single generation due to the length of the initial texts. Furthermore, 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 the corresponding synthetic entities to ensure consistency between the previously added PII and the new entities. The `Faker` library (Faraglia, 2014) was used to generate a diverse set of personal synthetic entities, located in six different countries.

After completion of this process, we obtained a dataset with personal information exclusively tied to the author of the text. In the final stage, we introduce non-PII entities that are not related to the author using the prompt in Figure 9.

4 Data Analysis

SPY's flexible pipeline for synthetic PII data generation demonstrates several 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 Figure 3 where the entities' positions are spread relatively uniformly across the texts, avoiding clustering in any specific section.

Entity	**Legal Questions**			**Medical Consultations**		
	pii 1	pii 2	final	pii 1	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	0.87	1.1 (**+0.23**)	0.75 (**-0.12**)	0.88	1.12 (**+0.24**)	0.89 (**+0.01**)
URL	1.07	1.34 (**+0.27**)	0.87 (**-0.2**)	1.03	1.32 (**+0.29**)	0.88 (**-0.15**)
Address	0.71	1.19 (**+0.48**)	0.87 (**+0.16**)	0.73	1.28 (**+0.55**)	1.06 (**+0.33**)
ID	0.39	0.98 (**+0.59**)	0.69 (**+0.3**)	0.53	1.05 (**+0.52**)	0.89 (**+0.36**)
avg.	0.79	1.14 (**+0.35**)	0.89 (**+0.1**)	0.81	1.17 (**+0.36**)	0.99 (**+0.18**)

Table 2: Frequency of entities calculated by dividing the total number of entities by the number of texts. Frequencies for each entity type are computed separately. *pii {k}* refers to the frequency of PII placeholders after k iterative updates using the prompt from Figure 8; *final* represents the frequency of PII entities after completing all stages of the pipeline from Figure 1.

Balanced Entity Counts: The number of entities by type is relatively balanced. For example, we observed that after running the pipeline, there were approximately 3,000–5,000 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 6.

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 *pii 2* there is a steady increase in the frequency of entities, calculated as the total number of entities divided by the number of texts. This flexibility in entity injection enables the generation of more entity-rich texts. We opted against more than two 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 previously generated PII placeholders, as shown in Table 2 in column *final*.

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 Experimental setup

5.1 Baselines

In the following, we outline several zero-shot baseline approaches we employ for PII detection.

Presidio (Microsoft, 2021) is a Microsoft SDK that provides a fast identification for PII entities by employing a combination of techniques including NER modules, regular expressions, and additional rule-based logic.

LLaMA-3-70B (AI@Meta, 2024) with zero-shot instruction to extract personal entities described in Figure 5. 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 Our approach

Our supervised solution is based on `DeBERTaV3-base` encoder (He et al., 2023). Fine-tuned `DeBERTa` encoder-based models have exhibited their capabilities in identifying named entities (Tirskikh and Konovalov, 2023). Since we do not divide the data into training and test sets, we evaluated the model in a domain-transfer scenario. Specifically, we train the DeBERTa model on data from one domain and assess its performance in another. The training hyperparameters can be found in Appendix A.

5.3 Evaluation Metrics

For our evaluation, we use precision, recall, and F1 score, which are standard metrics to assess token classification tasks (Sang and Meulder, 2003).

6 Experimental results

First, we verify that SPY contains a substantial amount of non-PII entities. To do this, we evaluated UniNER (Zhou et al., 2024) on the `name` entity type using the prompt shown in Table 4. The results indicate that Recall is significantly higher than Precision, suggesting that UniNER identified additional non-PII names. This observation is also supported by the example provided in Table 4.

Following the pipeline presented, we generated two datasets from the legal and medical domains. Table 3 shows how different models perform PII detection on the SPY dataset. We can clearly see that Presidio has a much lower Precision than the Recall for all the categories, meaning that it misclassified a large 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.

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

Table 3: 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** is LLaMA-3-70B zero-shot prompted LLM for PII task. **DeBERTa** is a model cross-validated on different domains of the SPY dataset. Blanks mean that entity class is not supported by the model.Presidio extracts addresses only at the geographical level, excluding street names and house numbers.

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

Table 4: UniNER evaluation results on the SPY dataset. Metrics are calculated specifically for `name` enity type, using prompts from the original UniNER paper (Zhou et al., 2024): *“What describes a person in the text?”*

DeBERTa validated on the 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. In general, encoder-based models have demonstrated their remarkable ability to transfer across tasks, domains, and languages (Karpov and Konovalov, 2023).

The encoder model specifically trained to detect

PII entities outperforms the 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.

7 Conclusions

In this study, we discuss 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 highlight 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 does not require human supervision. These advancements reinforce our commitment to safeguarding personal data, a significant area in today's digital landscape.

The generated dataset can be utilized to fine-tune the PII model independently or within the DeepPavlov framework (Savkin et al., 2024). To encourage research in the field, we make the SPY dataset freely available.

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 that is able to adapt 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.

Another significant limitation is that the generated PII entities only relate to 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.

Taking all the aforementioned factors into account, the trained model and generated dataset should not be used in a real production system to detect PII entities, anonymize documents, or be utilized in any other manner, except for research purposes.

Ethics Statement

While SPY methodology 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, all PII entities in the generated dataset are artificial.

We emphasize that the generated dataset and the methodology should be used only for research purposes.

We strongly discourage any use of our dataset that aims to undermine privacy protections or misuse the generated synthetic data for harmful purposes.

References

Griffin Adams, Alexander R. Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. From sparse to dense: GPT-4 summarization with chain of density prompting. *CoRR*, abs/2309.04269.

AI@Meta. 2024. Llama 3 model card.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoît Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 11829–11841. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *ACM CHIL '20: ACM Conference on Health, Inference, and Learning, Toronto, Ontario, Canada, April 2-4, 2020 [delayed]*, pages 214–221. ACM.

Dmitry Karpov and Vasily Konovalov. 2023. Knowledge transfer between tasks and languages in the multi-task encoder-agnostic transformer-based models. In *Computational Linguistics and Intellectual Technologies*, volume 2023.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2023. The stack: 3 TB of permissively licensed source code. *Trans. Mach. Learn. Res.*, 2023.

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, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason T. Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *Trans. Mach. Learn. Res.*, 2023.

Simona Mazzarino, Andrea Minieri, and Luca Gilli. 2023. NERPII: A python library to perform named entity recognition and generate personal identifiable information (short paper). 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 (AIxIA 2023), Rome, Italy, November 6th-7th, 2023*, volume 3551 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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

Arvind Narayanan and Vitaly Shmatikov. 2010. Myths and fallacies of "personally identifiable information". *Commun. 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. *Comput. 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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Maksim Savkin, Anastasia Voznyuk, Fedor Ignatov, Anna Korzanova, Dmitry Karpov, Alexander Popov, and Vasily Konovalov. 2024. DeepPavlov 1.0: Your gateway to advanced NLP models backed by transformers and transfer learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 465–474, Miami, Florida, USA. Association for Computational Linguistics.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *CoRR*, abs/2303.04360.

Danil Tirskikh and Vasily Konovalov. 2023. Zero-shot NER via extractive question answering. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VII*, pages 22–31, Cham. Springer Nature Switzerland.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. Gliner: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 5364–5376. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. Universalner: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A DeBERTa Hyperparameters

Hyperparameter	Value
Optimizer	AdamW
Adam β_1, β_2	0.9, 0.999
Adam ϵ	1e-6
Warm-up step	100
Context size	1,800
Learning rate (LR)	5e-6

Table 5: `DebertaV3-base` hyperparameters

B Data Analysis

Entity type	Domain	
	Legal questions	Medical Consultations
url	4,243	4,322
email	4,101	4,493
username	3,868	4,273
address	4,173	5,122
name	4,032	4,707
phone number	3,597	4,222
id_num	3,357	4,284

Table 6: Number of generated PII entities by type.

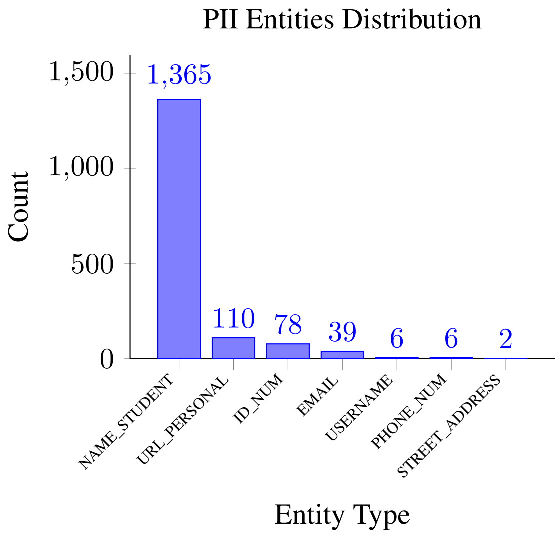


Figure 2: The distribution of entities present in the Kaggle PII dataset illustrates its highly imbalanced nature.

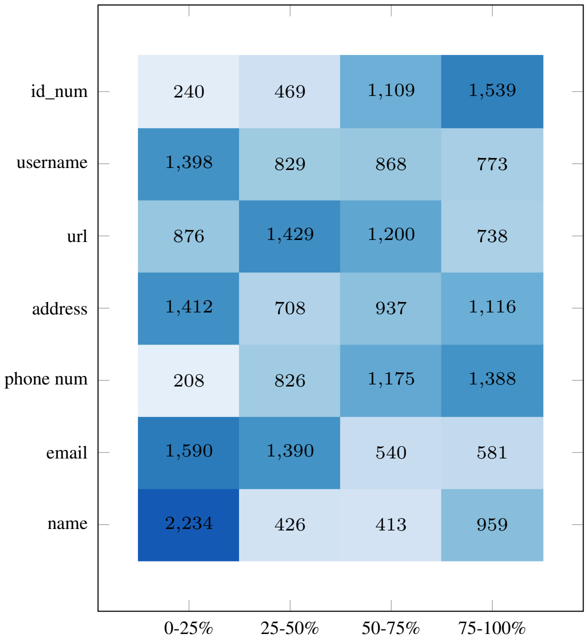
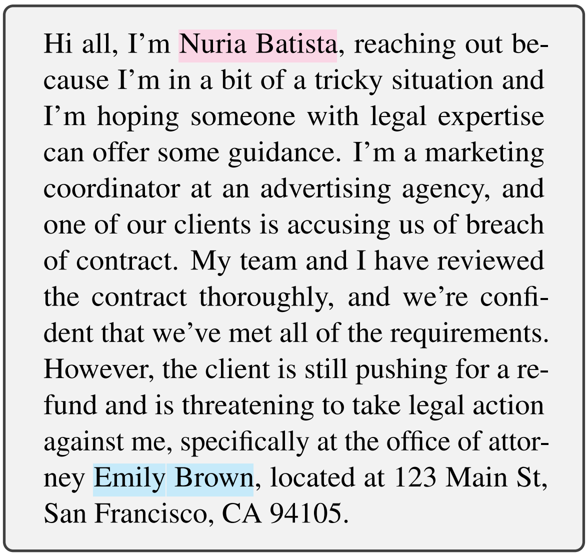


Figure 3: Heatmap showing the distribution of PII entity counts across relative position bins in the Legal Questions Domain of the SPY Dataset.

C UniNER "NAME" Class Prediction



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.

Figure 4: 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 Generation Pipeline 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 5: LLaMA-3-70B prompt for extracting PII entities from text.

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.

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 7: Prompt for biography generation. Placeholder “<generated-name-goes-here>” is replaced with random name.

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 8: 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 9: 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 10: Prompt for extracting PII from text.