Personally Identifiable Information Data Detection. NLP Course Project

Anna Marshalova, Olga Tikhobaeva, Timur Ionov

May 2024

Abstract

In this paper, we address the task of extracting personally identifiable information (PII) from educational data. In addition to the task of information extraction, our work also highlights the problem of quantity and quality of training data. We conducted experiments and compared optimization level and data level approaches to solve this problem: the first one is based on the use of Focal loss, and the second one is based on the use of a large language model (LLM) and Faker library for data augmentation. Our results show that both methods can achieve reasonable performance improvements, compared to the baseline. Our project code is available on github: `https://github.com/sir-timio/NLP_ods_course`.

1 Introduction

In the age of abundant educational data from various sources like ed tech, online learning, and research, the widespread presence of personally identifiable information (PII) in students writings poses a significant challenge. Analyzing and creating open datasets for educational advancement is hindered by the presence of PII, as releasing this data publicly could jeopardize student safety. Therefore, it is essential to thoroughly screen and clean educational data for PII before making it available to the public, a process that could be streamlined through the use of data science.

Currently, manually reviewing the entire dataset remains the most effective method for screening PII, but this approach comes with high costs and limits the scalability of educational datasets. While there are some traditional methods like regular expressions for automatic PII detection, these methods are most successful with PII that has common formats like emails and phone numbers. This type of PII detection systems often struggle to accurately identify names and differentiate between sensitive names, such as a student's name, and non-sensitive names, like a cited author.

In our work, we suggest an approach to solve the problem of extracting PII with the use of machine learning (NER models). This methods allow to extract any type of PII, including written in non-common formats. However, machine learning usage poses certain challenges, particularly concerning the quantity and quality of the training data. Therefore, in our study, we highlight methods to deal with class imbalance and lack of data on the optimization level by using Focal Loss, and on the data level through augmentation.

Our team consists of:

- **Anna Marshalova** - responsible for data generation and prompt tuning;
- **Olga Tikhobaeva** - responsible for exploratory data analysis (EDA) and literature review;
- **Timur Ionov** - responsible for setting up experiments and implementing training and evaluation.

2 Related Work

PII detection, often approached as a Named Entity Recognition (NER) task, encompasses a range of techniques tailored to different types of PII and data formats. This section provides an overview of the existing works on PII detection methods and techniques that can be used for synthetic PII data generation.

Traditional methods like regular expressions achieve fair accuracy in identifying phone numbers, email addresses [Aura et al., 2006], IP addresses [Allal et al., 2023] and other types of PII that have a strict and recognizable format. Regular expressions are particularly effective for structured tabular data.

Machine learning methods, on the other hand, offer versatility in detecting PII across various contexts, especially in unstructured text. ML algorithms, including Conditional Random Fields (CRF) [Minkov et al., 2005], Recurrent Neural Networks [Eder et al., 2020], and Transformer-based models [Johnson et al., 2020, Pilán et al., 2022, Li et al., 2023] are proficient in detecting names and addresses by learning from labeled datasets containing examples of PII instances, along with Faker [Faraglia, 2014] for synthetic PII generation.

Some approaches involve combining different methodologies for PII data detection, For instance, PII identification module of Presidio [Microsoft, 2021] leverages both regular expressions and NER models for primary data recognition, checksum for pattern validation, and keyword search to increase detection confidence.

However, the aforementioned machine learning methods require a large corpus of labeled data for training. When labeled data is limited or when entities are scarcely represented in the dataset, alternative approaches become necessary. Some works suggest using unsupervised methods, for example, [Islam et al., 2023] employs various outlier detection algorithms for PII detection in unstructured data. Other viable solutions are automatic data annotation or enrichment of the labeled dataset with synthetic data.

Methods for automatic entity labeling involve using large language models (LLMs) for few-shot NER. For example, PromptNER [Ashok and Lipton, 2023] operates by prompting an LLM to generate a list of potential entities in a given text along with corresponding explanations that validate their compatibility with the provided entity type definitions.

NuNER [Bogdanov et al., 2024] diverges from direct annotation of single-domain datasets for specific NER tasks, using LLMs to annotate a multi-domain dataset for a variety of NER problems. Its architecture comprises two distinct sub-networks: the first encodes the input text, while the second encodes the entity type.

As for synthetic data generation, some works [Tang et al., 2023] advocate for using a small set of human-labeled examples to guide LLMs in generating diverse synthetic data with varying sentence structures and linguistic patterns. Their method, followed by a post-processing step to filter out low-quality or duplicated samples, enhances the quality and diversity of synthetic data, making it suitable for fine-tuning local pre-trained language models.

Another framework for data augmentation for low-resource NER is BioAug [Ghosh et al., 2023], which involves masking out the majority of tokens, except named entities and their relations, and training BART model [Lewis et al., 2020] on a text reconstruction task. Such an approach ensures that entities are placed in the right context and augmentations remain factually correct.

Lastly, NERPII [Mazzarino et al., 2023], a structured data pseudonymization library, utilizes Presidio [Microsoft, 2021] and a BERT model [Devlin et al., 2019] for PII entity recognition.

In summary, PII detection methods range from traditional techniques like regular expressions to machine learning models, often requiring labeled data. In low-resource scenarios, unsupervised methods, LLM-based entity labeling, and synthetic data generation strategies are employed to overcome annotation scarcity challenges.

3 Dataset

The original data source for training and testing our approaches is the dataset provided in the competition The Learning Agency Lab - PII Data Detection[1]. on the platform Kaggle. The dataset for the competition consists of around 22,000 essays that were penned by students participating in a massively open online course. These essays were all written in response to a single assignment prompt, in which students were tasked with applying course concepts to a real-world issue.

To safeguard the privacy of the students, the original PII present in the dataset has been substituted with similar surrogate identifiers through a partially automated method. The majority of the essays have been allocated to the test set (70%), prompting competitors to utilize publicly available external datasets to enhance their training data.

[1] https://www.kaggle.com/competitions/pii-detection-removal-from-educational-data/data

The competition asks competitors to assign labels to the following seven types of PII:

- NAME_STUDENT - The full or partial name of a student that is not necessarily the author of the essay. This excludes instructors, authors, and other person names.
- EMAIL - A student's email address.
- USERNAME - A student's username on any platform.
- ID_NUM - A number or sequence of characters that could be used to identify a student, such as a student ID or a social security number.
- PHONE_NUM - A phone number associated with a student.
- URL_PERSONAL - A URL that might be used to identify a student.
- STREET_ADDRESS - A full or partial street address that is associated with the student, such as their home address.

The data includes a document identifier, the full text of the essay, a list of tokens, information about whitespace, and token annotations:

- (int): the index of the essay
- document (int): an integer ID of the essay
- full_text (string): a UTF-8 representation of the essay
- tokens (list) (string): a string representation of each token. The documents were tokenized using the SpaCy English tokenizer.
- trailing_whitespace (list) (bool): a boolean value indicating whether each token is followed by whitespace (True - if is followed, False - if not).
- labels (list) [training data only] (string): a token label in BIO format

The test set is hidden from the competitors, so in this overview, we will only show the statistics for the train set. Base statistics for the train set are presented in Table 1.

Unit	**number**
`all documents`	6807
`documents without entities`	5862

Table 1: Dataset base statistics

Statistics for the labels in train set are presented in Table 2.

As can be seen from the statistics presented, more than 85% of documents do not contain PII at all. Furthermore, the essays are sparsely populated with

Labels	**Number of labels in train**	**Number of docs with labels**
B-NAME_STUDENT	1365	891
I-NAME_STUDENT	1096	814
B-URL_PERSONAL	110	72
I-URL_PERSONAL	1	1
B-ID_NUM	78	33
I-ID_NUM	1	1
B-EMAIL	39	24
I-EMAIL	0	0
B-STREET_ADDRESS	2	2
B-STREET_ADDRESS	20	2
B-PHONE_NUM	6	4
I-PHONE_NUM	15	3
B-USERNAME	6	5
I-USERNAME	0	0
O	4989794	6807

Table 2: Dataset statistics for labels

entities, and some entities are so rare that they appear only once or twice across the entire dataset (e.g., B-STREET_ADDRESS). All of this complicates the process of NER models training and requires additional work to deal with the strong class imbalance or to complete the dataset with new data. Our solutions to this problem will be presented in the following sections.

A more detailed exploratory data analysis is available in our project repository[2].

4 Model Description

In our experiments, we used the following stack to solve both the PII extraction problem, directly, and the training data imbalance problem:

- DeBERTa as a baseline NER model;
- Faker for fake PII generation;
- Mixtral-7Bx8 for smooth PII insertion into texts.

4.1 DeBERTa

The DeBERTa model was introduced in [He et al., 2020]. The authors proposed a new model architecture that improves the BERT [Devlin et al., 2019] and RoBERTa [Liu et al., 2019] models using two novel techniques.

The first one is the disentangled attention mechanism, where every word is represented using two vectors that encode its content and position separately.

[2] https://github.com/sir-timio/NLP_ods_course/blob/main/pybooks/eda.ipynb

Additionally, the attention computation algorithm is adjusted to explicitly consider the connections between the content and positions of tokens. For illustration, the words “research” and “paper” exhibit stronger dependence when they are in close proximity than when they are in distant sections of the text. This example clearly demonstrates the importance of considering content-to-position relations.

Second, there is an improved mask decoder (EMD) utilized to substitute the output softmax layer for the prediction of the masked tokens during model pretraining. As explained by the authors, EMD consists of two input blocks:

- H — the hidden states from the previous Transformer layer.
- I — any necessary information for decoding (e.g. hidden states H, absolute position embedding or output from the previous EMD layer).

The schema of EMD block is presented in Figure 1.

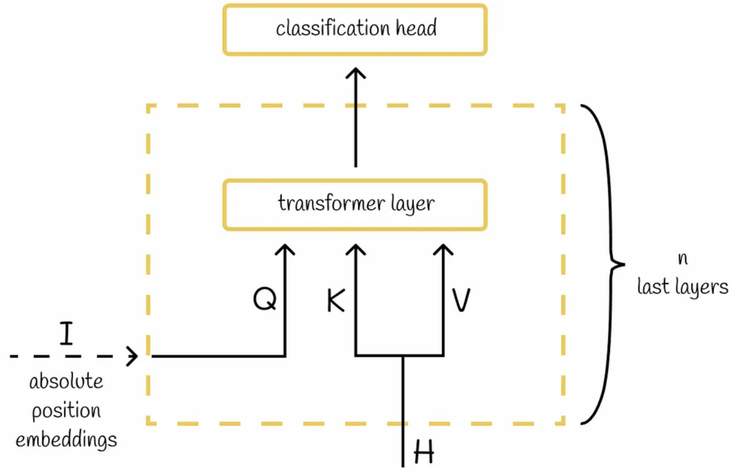

Figure 1: Enhanced mask decoder in DeBERTa

In general, a model may contain several EMD blocks. In such cases, these blocks are configured based on the following guidelines:

- the output of each EMD layer is the input I for the next EMD layer;
- the output of the last EMD layer is fed to the language model head.

In the case of DeBERTa, the number of EMD layers is set to n = 2 with the position embedding used for I in the first EMD layer.

In our experiments we used DeBERTaV3[3], which further improved the efficiency of DeBERTa using ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing.

[3] https://huggingface.co/microsoft/deberta-v3-base

4.2 Faker

Faker[Faraglia, 2014] is a Python library for fake data generation. It is able to generate different types of PII, including names, addresses, phone numbers, email addresses, etc. Faker. A key feature of Faker is its ability to customize the generated entities to fit specific regions, which is crucial for projects involving international data such as our collection of essays from diverse geographical backgrounds.

4.3 Mixtral-7Bx8

Mixtral-7Bx8 [Jiang et al., 2024], is a Sparse Mixture of Experts (SMoE) language model. It adopts the transformer architecture but distinguishes itself by organizing each layer into 8 feedforward blocks called 'experts.' At each step, a router network selects two experts to process each token, subsequently combining their outputs. Despite the potential access of each token to 47 billion parameters, the model effectively utilizes only 13 billion during inference. Due to its efficient compute management, Mixtral outperforms larger models while requiring significantly less resources. For our experiments, we employed Mixtral-8x7B-Instruct-v0.1-GPTQ[4], which is a version of Mixtral-8x7B fine-tuned to follow instructions and quantized with GPTQ [Frantar et al., 2022] method.

5 Experiments

In this section, we detail the empirical setup and methodologies employed in our study. The primary challenge in training a model for PII extraction is the insufficient amount of training data, particularly the underrepresentation of certain PII types. To address this issue, we implemented two main strategies aimed at enhancing model performance, each tackling the imbalance problem from different angles:

- **Optimization-Level Adjustment Using Focal Loss:** We adapted the standard Cross-Entropy loss into Focal Loss to better handle imbalanced datasets. This method modifies the focus of the training process by increasing the importance of misclassified or difficult-to-classify examples, thereby ensuring that the model does not overlook the less frequent classes. This approach allowed us to optimize model performance using only the data available from the competition, without the need for external data sources.

- **Data-Level Enhancement Through Augmentation:** In contrast to the optimization-level approach, we expanded our dataset using a combination of LLM (Mixtral-7Bx8) for realistic text generation and Faker for synthetic PII creation. By artificially enhancing the dataset with a higher

[4] `https://huggingface.co/TheBloke/Mixtral-8x7B-Instruct-v0.1-GPTQ`

incidence of PII, this method directly addresses the root of the imbalance by increasing the volume and variety of training examples. This not only improves the model's exposure to different PII types but also aids in generalizing better across unseen data.

These approaches are fundamentally complementary but operate on different levels of the training pipeline. By implementing and comparing these methods, we aim to not only assess their individual impacts on model performance but also explore their potential synergies in creating a robust system for PII detection. The results of these experiments will provide insights into the effectiveness of tackling dataset imbalance from both the optimization and data perspectives.

5.1 Focal loss

One of the approaches we tried to resolve the label sparsity problem, was the incorporation of focal loss [Lin et al., 2017] into the base pipeline. This loss is a modification of the standard Cross-Entropy loss, designed for handling imbalanced datasets. It dynamically adjusts the weights assigned to different classes during training, focusing more on hard-to-classify examples. Mathematically, focal loss can be represented as:

$$\text{Focal Loss} = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{1}$$

In this formula, p_t represents the predicted probability of the correct class, α_t is a weighting factor used to balance the loss for each class, and γ is a focusing parameter that adjusts the rate at which easy examples are down-weighted.

5.2 Data augmentation

To address the challenge of rare and unevenly distributed entities within a dataset, we employed a data augmentation technique using large language models (LLMs) to artificially insert fake PII entities into existing text passages. This method leverages the linguistic capabilities of LLMs to generate contextually appropriate sentences that include specific entity types, thereby enriching the dataset with diverse instances of rare entities. This method systematically incorporates synthetic entities, ensuring diverse and comprehensive coverage for model training. Here is a breakdown of each step:

5.2.1 Fake entities generation

The first step in our data augmentation process involves generating synthetic entities using the Faker library and some rule-based algorithms. Provided that the dataset includes essays from students of various nationalities, it was essential to tailor the synthetic PII to reflect the diversity of the student population. To achieve this, we utilized Faker's capabilities to generate PII that is regionally appropriate for students from different countries 3.

	fr_FR	en_US
ID	8534955202OgSE	603335162zevWU
Name	Marine Baudry	Jennifer Key
Email	ysimon@example.org	jennifer_key@bell-jones.info
Username	caronjuliette	uestes
Phone	0469986921	4788039896
URL	https://github.com/caronjuliette	https://instagram.com/uestes
Address	3, chemin Lévy 32895 Masse	PSC 0864, Box 3570 APO AE 06503

Table 3: Examples of region-specific synthetic PII generated using Faker

5.2.2 Entities insertion with LLM

In our project, we employed a text rewriting strategy to integrate PII into existing essays using LLMs. This process aimed to insert PII in a manner that preserved the natural flow and integrity of the text. The initial approach involved directly generating text with embedded fake PII using LLMs, but this method encountered several challenges:

1. **Unnatural text flow:** Direct generation often resulted in essays where the inserted PIIs felt forced or intrusive, disrupting the natural tone and coherence of the text.

2. **Loss and corruption of entities:** There was a significant issue with some PIIs being omitted or corrupted during generation, leading to incomplete representation of the required personal information.

To overcome these difficulties, we use real-world essay without PII and using "dummy" entities that easy to put into text without loss1.

```
<ESSAY>
Task description: You are given a text that lacks
   specific personal information identifiers (PII).
   Your task is to rewrite the above essay by adding
   all the personal information listed below, while
   preserving the naturalness and integrity of the
   text.

<DUMMY PII COMBO>

Requirements:
- Make sure to mention all of the examples of
   personal information in generated essay.
- Add entity in such a way that it seamlessly
   integrates into the context and does not seem
   overly intrusive or out of place.
- Distribute personal information evenly throughout
   the essay, do not stack it all in one place.
```

Listing 1: Prompt used for LLM-based entity insertion

After the process of inserting dummy entities using the LLM, it is crucial to ensure that the entities are correctly and fully integrated into the text. This next phase involves a meticulous review and validation process, followed by replacing the dummy entities with actual data generated from Faker.

5.3 Metrics

To effectively evaluate the performance of our model, we used the following metrics: Precision, Recall, F1 Score, and the F5 Score. These metrics are particularly important in scenarios where class imbalance may affect the performance of the model, such as in our dataset of student essays with sparse entity occurrences.

Precision measures the accuracy of positive predictions. Formally, it is defined as the ratio of true positives to the sum of true and false positives:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall measures the ability of the model to detect all relevant instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1 Score is the harmonic mean of Precision and Recall, providing a balance between the two. It is especially useful when you need to balance the precision and recall performance measures:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

F5 Score is a variation of the F-measure that places a higher emphasis on recall compared to precision. It is particularly useful in situations where failing to detect a true instance has a higher cost than incorrectly detecting an instance as true. Also, this metric provided as main for competition host and available for hidden test set. The F5 Score is calculated by the following formula:

$$F5 = (1 + 5^2) \cdot \frac{Precision \cdot Recall}{(5^2 \cdot Precision) + Recall} \tag{5}$$

These metrics collectively provide a comprehensive overview of model performance, emphasizing different aspects of prediction accuracy and robustness in handling class imbalances.

5.4 Experiment Setup

The dataset was divided into training and validation sets using a stratified split of 75/25 percent.

- Batch Size: 2
- Optimizer: Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Train Steps: 10000
- Max tokens length: 2300
- Learning rate: 1×10^{-5}
- Weight decay: 0.01
- Warmup Steps: 600

In focal loss were used default parameters $\alpha = 1$ and $\gamma = 2$.

In the case of data augmentation, it was generated 3000 samples and selected 2250 after sanity check. More detailed exploratory data analysis is available in our project repository[5].

5.5 Baselines

In our experiments, we established a strong baseline model to benchmark the performance of our proposed enhancements. The baseline chosen for this task was the DeBERTa-base model, paired with the traditional Cross-Entropy loss and a downsampling strategy for balancing the dataset.

- **DeBERTa Model:** We opted for DeBERTa due to its advanced architecture that effectively handles long context dependencies within text. This capability is crucial for our task, as the essays presented in our dataset necessitate the understanding and incorporation of extended textual contexts to accurately detect and classify PII.
- **Dataset Downsampling:** To address the imbalance between texts containing PII and those without, as baseline, we implemented a downsampling strategy. Texts without PII were downsampled to achieve a ratio of 1:3 (PII to non-PII texts).

[5] https://github.com/sir-timio/NLP_ods_course/blob/main/pybooks/generated.ipynb

6 Results

Model Configuration	**Precision**	**Recall**	**F1**	**F5**	**Test F5**
Baseline	0.877	0.765	0.817	0.768	0.847
+ Focal	0.885	0.768	0.822	0.772	0.853
+ Augmentation	0.871	**0.813**	0.841	**0.815**	**0.897**
+ Focal + Augmentation	**0.897**	0.809	**0.851**	0.812	0.887

Table 4: Performance metrics of models with various configurations, validation and hidden test.

6.1 Impact of focal loss

The introduction of focal loss increase precision from 0.877 to 0.885 on the validation set. This suggests that focal loss helps the model make more confident predictions about the classes, effectively reducing the false positive rate. However, the influence of focal loss on the F5-score, which is heavily influenced by recall, is negligible. This indicates that while the model becomes more selective and accurate in its predictions, it does not necessarily capture more positive cases overall, especially in imbalanced datasets where rare classes are critical.

6.2 Impact of data augmentation

The introduction of generated samples leads to a significant improvement in recall from 0.765 to 0.813. This increase in recall contributes to the substantial gains seen in the F5-score on both the validation and hidden test sets (from 0.847 to 0.897). Data augmentation addresses the core challenge of dataset imbalance by enriching the training data with more instances of underrepresented classes, thereby enabling the model to detect a broader range of PII instances.

6.3 Impact of combination approaches

When combining focal loss and data augmentation, the precision reaches the highest value of 0.897, and the F1-score also peaks at 0.851, demonstrating that this combination effectively enhances overall model accuracy and balance between precision and recall. However, despite these gains, the F5-score does not see parallel improvements, indicating a trade-off where enhanced precision may come at the expense of recall in certain contexts.

7 Conclusion

In this study, we addressed the challenge of detecting personally identifiable information in educational texts, exploring strategies of overcoming dataset imbalance and improving model performance. We leveraged two techniques to enhance the accuracy and robustness of our PII extraction model.

Firstly, we tried using Focal Loss instead of standard Cross-Entropy loss. Its dynamic adjustment, which prioritizes misclassified examples, enhanced the model's ability to recognize less frequent PII types. This optimization-level improvement demonstrated effectiveness in boosting performance without the need for extra data sources.

Secondly, we employed data augmentation through LLM (Mixtral-7Bx8) and Faker, effectively expanding the dataset with realistic and synthetic PII instances. This approach directly addressed the imbalance by increasing the volume and diversity of training examples, ultimately enhancing the model's exposure to various PII types and its ability to generalize across unseen data. As a result, this approach led to notable improvements in model performance.

The combination of focal loss and data augmentation yielded the most substantial enhancement, achieving the highest precision and F1 scores among all model configurations. This synergy highlights the complementary nature of optimization-level adjustments and data-level enhancements in addressing dataset imbalance and enhancing model performance.

Overall, our findings underscore the effectiveness of a multi-faceted approach to PII detection, combining algorithmic optimizations with data-driven strategies to create more accurate and reliable models. By addressing the inherent challenges of dataset imbalance, our methodologies contribute to advancing the field of educational data analysis while upholding privacy and security standards in handling sensitive information.

8 Future work

Moving forward, future research directions could explore additional techniques for mitigating dataset imbalance, such semi-supervised learning approaches. Additionally, we aim to stabilize end-to-end generation processes to ensure more consistent and reliable data generation. This improvement could facilitate the deployment of our methodologies in real-world applications more effectively. Furthermore, the generation techniques developed in this project could be adapted for other tasks or languages that suffer from low-resource or rare labels challenges.

References

[Allal et al., 2023] Allal, L. B., Li, R., Kocetkov, D., Mou, C., Akiki, C., Ferrandis, C. M., Muennighoff, N., Mishra, M., Gu, A., Dey, M., et al. (2023). Santacoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*.

[Ashok and Lipton, 2023] Ashok, D. and Lipton, Z. C. (2023). Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.

[Aura et al., 2006] Aura, T., Kuhn, T. A., and Roe, M. (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.

[Bogdanov et al., 2024] Bogdanov, S., Constantin, A., Bernard, T., Crabbé, B., and Bernard, E. (2024). Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv preprint arXiv:2402.15343*.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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.

[Eder et al., 2020] Eder, E., Krieg-Holz, U., and Hahn, U. (2020). CodE alltag 2.0 — a pseudonymized German-language email corpus. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4466–4477, Marseille, France. European Language Resources Association.

[Faraglia, 2014] Faraglia, D. (2014). Faker. `https://github.com/joke2k/faker`.

[Frantar et al., 2022] Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. (2022). Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

[Ghosh et al., 2023] Ghosh, S., Tyagi, U., Kumar, S., and Manocha, D. (2023). Bioaug: Conditional generation based data augmentation for low-resource biomedical ner. SIGIR '23, page 1853–1858, New York, NY, USA. Association for Computing Machinery.

[He et al., 2020] He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

[Islam et al., 2023] Islam, M. R., Kayem, A. V., and Meinel, C. (2023). Enabling pii discovery in textual data via outlier detection. In *International Conference on Database and Expert Systems Applications*, pages 209–216. Springer.

[Jiang et al., 2024] Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. (2024). Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

[Johnson et al., 2020] Johnson, A. E., Bulgarelli, L., and Pollard, T. J. (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.

[Lewis et al., 2020] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

[Li et al., 2023] Li, R., Allal, L. B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J., et al. (2023). Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

[Lin et al., 2017] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Mazzarino et al., 2023] Mazzarino, S., Minieri, A., and Gilli, L. (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] Microsoft (2021). Presidio. `https://microsoft.github.io/presidio/`.

[Minkov et al., 2005] Minkov, E., Wang, R. C., and Cohen, W. (2005). Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 443–450.

[Pilán et al., 2022] Pilán, I., Lison, P., Øvrelid, L., Papadopoulou, A., Sánchez, D., and Batet, M. (2022). The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101.

[Tang et al., 2023] Tang, R., Han, X., Jiang, X., and Hu, X. (2023). Does synthetic data generation of llms help clinical text mining?