XCS224U Final Project Report*

🎭 *Academic Anonymity: Advancing PII Detection in Essays through Longformer*

Jonathan Algar
Lisbon, Portugal
jonathan.algar@gmail.com

*Updated since submission.
Last update: May 24, 2024.

Abstract

This project concerns the task of detecting Personally Identifiable Information (PII) in a domain-specific text corpus. The experiments in this project use student essays as the domain.

Microsoft's Presidio SDK (Mendels et al., 2018) has established itself as the default choice for industry practitioners for the task of detecting PII, irrespective of the domain (see, for example, Aziz and Straiton 2023). This choice is especially true since Presidio's recent integration with popular practitioner tools such as LangChain.

This project shows that a Longformer model (Beltagy et al., 2020) fine-tuned on a high-quality, human-annotated, domain-specific dataset yields considerable outperformance in the identification of PII ($F_5 = 0.831$) over the Presidio benchmark ($F_5 = 0.735$).

Furthermore, I show that synthetically generated data can be highly effective in supplementing a train set to bolster overall performance ($F_5 = 0.936$).

The result supports the long-standing conclusion of Chen et al. 2015 (summarized by Hathurusinghe et al. 2021) that for the training of a robust PII recognizer, a customized domain-specific annotated dataset is needed.

I conclude by discussing potential directions for further research and novel industry applications based on the results of these experiments.

1 Hypothesis

Microsoft's Presidio SDK (Mendels et al., 2018) has established itself as the default choice for industry practitioners for the task of detecting PII, irrespective of the domain (see, for example, Aziz and Straiton 2023). This choice is especially true since Presidio's recent integration with popular practitioner tools such as LangChain. The recognition engine uses a combination of techniques to detect PII entities: regular expressions, NER using spaCy, and predefined context words.

I will apply the SDK to the test set of a novel domain-specific dataset: student essays.

I hypothesize that the benchmark will be significantly outperformed, as measured by standard classifier metrics, by a custom model fine-tuned on the train set of a high-quality, human-annotated, domain-specific dataset.

If the primary hypothesis holds, it would add a further data point in support of the conclusion of Chen et al. 2015 (summarized by Hathurusinghe et al. 2021) that for the training of a robust PII recognizer, a customized domain-specific annotated dataset is needed.

The secondary hypothesis is that the primary hypothesis holds principally because of *in-context* PII, which is highly domain-specific.

2 Prior literature

The main problem addressed by the related published papers I assessed is how to use a corpus containing sensitive data for machine learning and downstream NLP tasks while protecting privacy. This is an important problem because:

- Clinical text, for example, contains a wealth of valuable information that can be used to train machine-learning models for various healthcare applications. However, this data also contains sensitive personal information about patients that must be kept private.

- Regulations like The Health Insurance Portability and Accountability Act of 1996 (HIPAA) in the United States and General Data Protection Regulation (GDPR) in the European Union place strict requirements on the handling of personal information. De-identification is often necessary for data to

be used for research purposes in compliance with these regulations.

- At the same time, over-zealous de-identification can destroy the utility of the data for machine learning. It is important to advance methods that protect privacy while preserving as much useful information as possible.

Since around 2020 the machine learning literature been unanimous in using transformers as the baseline for identifying PII in domain-specific corpora, irrespective of the domain or structure of the underlying text.

The domain of the underlying corpora varies across the related published papers I assessed. For example, while Hathurusinghe et al. 2021 and Vakili et al. 2022 focus on the medical domain, Pilán et al. 2022 focus on legal cases, and van der Plas 2022 on code.

Pilán et al. 2022 is the key paper for the intrinsic evaluation of text anonymization methods. The paper presents the Text Anonymization Benchmark (TAB), an annotated corpus and associated set of evaluation metrics designed to assess the performance of text anonymization methods.

The authors found a Longformer model fine-tuned on the TAB train set achieved superior performance across all the metrics against two baseline methods: Neural NER (RoBERTa)[1] and Presidio[2].

3 Data

The data used for the experiments in this paper is a corpus of 22,000 academic essays written by students enrolled in a Massive Open Online Course (MOOC) at Vanderbilt University in response to a single question.

This novel data was distributed in the context of a Kaggle competition (Holmes et al., 2024). The publicly distributed version of the dataset skews 70% to the *unannotated* test set versus 30% annotated for train and dev. For the experiment I advance in this project on the publicly distributed version of the data, I will focus on the 30% of annotated essays and split these 6,807 samples 80% for train, 10% for dev, and 10% test.

[1] *"A neural NER model based on the RoBERTa language model (Liu et al. 2019) and fine-tuned for NER on Ontonotes v5 (Weischedel et al. 2011), as implemented in spaCy."*

[2] *"We provide evaluation results for Presidio under two configuration settings, namely, the default mode and one in which the detection of organization names (governments, public administration, companies, etc.) is also activated."*

The data is distributed as a `json` (Holmes et al., 2024) and contains the following keys for each student essay:

- `document`: An integer ID uniquely identifying the essay.
- `full_text`: The complete text of the essay as a UTF-8 string. Includes all of the essay's content as it was submitted without modification.
- `tokens`: A list of strings, each representing a tokenized portion of the `full_text`. The SpaCy tokenizer (Honnibal et al., 2020) was used for tokenization.
- `trailing_whitespace`: A list of boolean values corresponding to each token in the `tokens` list. Each boolean indicates whether the token is followed by whitespace in the `full_text`.
- `labels`: A list of labels in the BIO (Beginning, Inner, Outer) format, corresponding to each token in the `tokens` list. A label with a prefix of `B-` indicates the beginning of a PII entity, `I-` signifies a continuation of a PII entity, and `O` indicates the token is not considered PII. This labeling schema facilitates precise identification and categorization of PII within the essays. See Figure 1 for a visual example of the labeling of the top part of one essay's `full_text`.

The essays contain the following types of PII entities:

- `NAME_STUDENT`: The full or partial name of a student. This type includes only the names of students and excludes names of instructors, authors, or any other individuals.
- `EMAIL`: The email address associated with a student.
- `USERNAME`: Any username a student uses which could provide hints to their identity or activities.
- `ID_NUM`: A unique number or sequence of characters assigned to a student, such as a student ID or social security number, that can be used for identification.

- PHONE_NUM: A contact phone number belonging to a student.
- URL_PERSONAL: A URL that is personally associated with a student. Includes blogs, portfolios, or profiles.
- STREET_ADDRESS: The full or partial physical address of a student. This includes any information that could be used to physically locate a student.

3.1 Analysis of original train set

Table 1 shows only a minority of essays in the originally constructed train set contain any PII.

Table 1: Distribution of essays with and without PII in the original train set.

Category	Count	Percentage
Essays with PII	763	14%
Essays without PII	4682	86%
Total essays	5445	100%

And Figure 2 shows the distribution of PII is heavily skewed to student names. Indeed, for several BIO labels, there are very few examples.

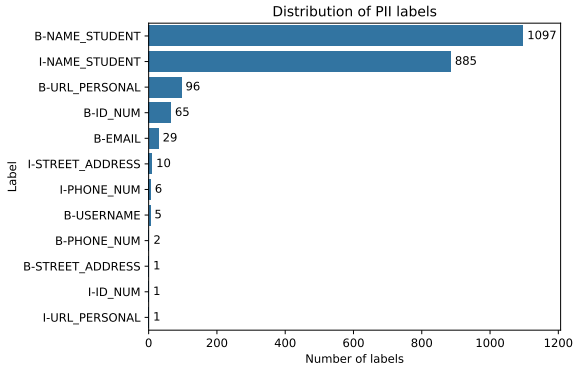


Figure 2: Original train set.

3.2 Synthetic data

To address the limited number of examples of a subset of the BIO labels in the original train set, I augment it with 1000 synthetically generated labeled essays.

To achieve this, I first use the Faker library (Faraglia and Contributors, 2023) to generate 1000 sets of PII data. An example set:

```
{
  'ID_NUM': '739916633Gmhl3hvcXqDgCIug',
  'NAME_STUDENT': 'Bryan Evans',
  'EMAIL': 'bryan.evans@gonzalez.com',
  'USERNAME': 'bryan.evans5',
  'PHONE_NUM': '+1-876-799-4028x1242',
  'URL_PERSONAL': 'https://youtube.com/c/bryan.
     evans5',
  'STREET_ADDRESS': '584 Patrick Hollow Apt. 760
     North Keith, DE 20952'
}
```

Based on the observed patterns of PII in the essays in the train set, I constructed two prompts.

PROMPT_1:

```
{
  "role": "system",
  "content": "You are a student tasked with
   writing an essay on how you applied a
   specific design thinking tool to address a
   challenge or problem in your life. In
   addition, you will be given a JSON
   containing your personal information: ID_NUM
   , NAME_STUDENT, EMAIL, USERNAME, PHONE_NUM,
   URL_PERSONAL, STREET_ADDRESS. Include all
   this information in the essay in a suitable
   format at the beginning of the essay."
},
{
  "role": "user",
  "content": "Your personal information: " +
   str(pii_set)
}
```

PROMPT_2:

```
{
  "role": "system",
  "content": "You are a student tasked with
   writing an essay on how you applied a
   specific design thinking tool to address a
   challenge or problem in your life. In
   addition, you will be given a JSON
   containing your personal information: ID_NUM
   , NAME_STUDENT, EMAIL, USERNAME, PHONE_NUM,
   URL_PERSONAL, STREET_ADDRESS. Weave all this
    personal information into the essay."
},
{
  "role": "user",
  "content": "Your personal information: " +
   str(pii_set)
}
```

50% of the set of generated PII data were allocated to PROMPT_1 and 50% to PROMPT_2. I generated the essays using OpenAI's `gpt-3.5-turbo-0125` model with `temperature=0.7`. Post-processing was then done to construct a dataset congruent with the original train set. This included tokenizing the essays using the SpaCy tokenizer (Honnibal et al., 2020), generating the `trailing_whitespace` boolean list, and assigning the appropriate BIO labels.

3.2.1 Analysis of synthetic data

And Figure 3 shows a far more balanced distribution of PII labels relative to the original train set.

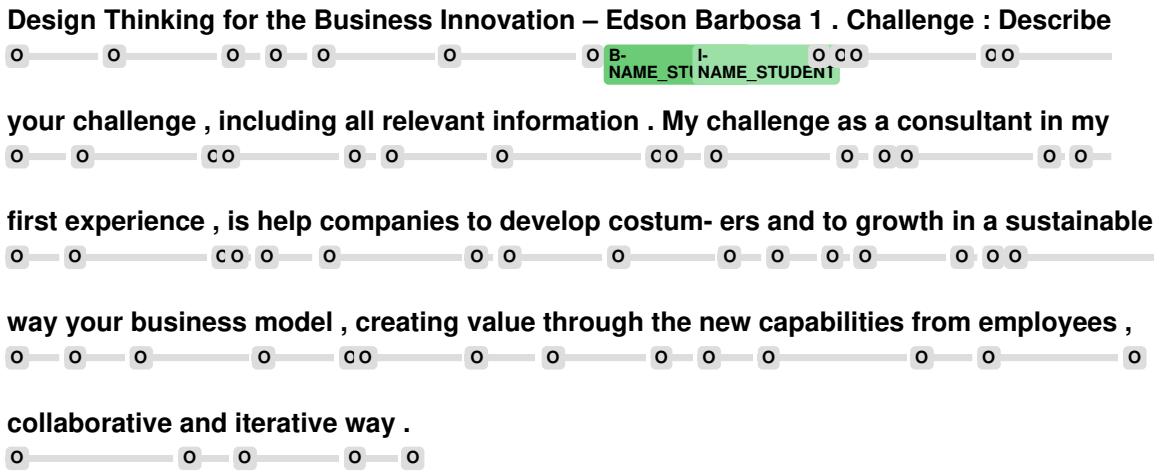


Figure 1: Using `displaCy` to visualize the BIO labels associated with the tokens of the first two sentences of a randomly selected essay containing PII in the original train set.

Table 2: Distribution of essays with and without PII in the generated synthetic dataset.

Category	Count	Percentage
Essays with PII	948	94.8%
Essays without PII	52	5.2%
Total essays	1000	100%

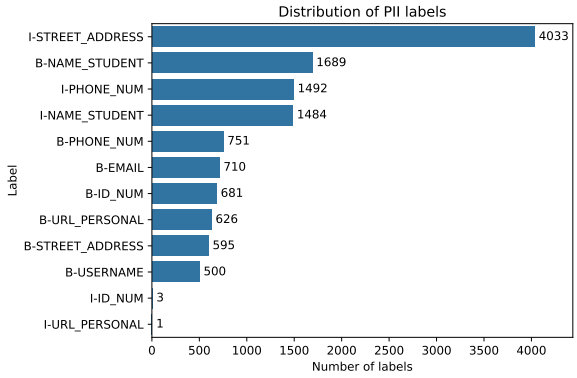


Figure 3: Train set with synthetic data.

4 Model

4.1 High-level overview

The Longformer (Beltagy et al., 2020) is a BERT-style transformer designed to efficiently process long documents using a combination of sliding window local attention and global attention. It is built on top of the RoBERTa checkpoint and was pre-trained on long documents using a masked language modeling objective. Pilán et al. 2022 showed it is a strong base model to use for the task of identifying PII.

The full architecture of the model used for this paper's experiments:

0. `allenai/longformer-base-4096` to process the tokenized input essays and produce contextualized token representations.

1. A bidirectional long short-term memory (LSTM) layer to further refine the token representations and capture sequential dependencies.

2. A linear output layer to project the LSTM hidden states onto the space of BIO labels.

3. A softmax activation function applied to the output layer to obtain probability distributions over the BIO labels for each token.

```
INFO:lightning.pytorch.callbacks.model_summary:
  | Name               | Type            | Params
-------------------------------------------------------
0 | transformers_model | LongformerModel | 148 M
1 | head               | LSTMHead        | 3.5 M
2 | output             | Linear          | 10.0 K
3 | loss_function      | CrossEntropyLoss| 0
-------------------------------------------------------
152 M     Trainable params
0         Non-trainable params
152 M     Total params
608.858   Total estimated model params size (MB)
```

The model was fine-tuned on two datasets independently: (1) the original train set and (2) the original train set augmented with the synthetically generated data. The objective was to minimize the cross-entropy loss between the predicted BIO label probabilities and the human-annotated gold BIO

labels. To address class imbalance, essays without PII were downsampled during training.

During training the model was optimized using the `AdamW` optimizer with a learning rate of 1×10^{-5} and a cosine learning rate scheduler. Training was for a maximum of four epochs with five-fold stratified cross-validation.

4.2 Mathematical summary

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be the input sequence of tokens for a given student essay. The Longformer computes a contextualized representation $\mathbf{h} = (h_1, \ldots, h_n)$, where $h_i \in \mathbb{R}^d$ ($d = 768$) is the hidden state corresponding to token x_i. The Longformer uses a combination of local and global attention to efficiently process long sequences:

$$\mathbf{h}^{(l)} = \texttt{LocalAttn}(\mathbf{h}^{(l-1)}, \mathbf{W}_Q^{(l)}, \mathbf{W}_K^{(l)}, \mathbf{W}_V^{(l)})$$

$$\mathbf{h}^{(g)} = \texttt{GlobalAttn}(\mathbf{h}^{(l)}, \mathbf{W}_Q^{(g)}, \mathbf{W}_K^{(g)}, \mathbf{W}_V^{(g)})$$

where $\mathbf{W}_Q^{(l)}, \mathbf{W}_K^{(l)}, \mathbf{W}_V^{(l)}, \mathbf{W}_Q^{(g)}, \mathbf{W}_K^{(g)}, \mathbf{W}_V^{(g)}$ are learnable parameters for the local and global attention mechanisms, respectively. The local attention is computed within a fixed-size window around each token, while the global attention allows a subset of tokens to attend to all other tokens in the sequence.

The LSTM layer further processes the hidden states to produce refined representations $\mathbf{r} = (r_1, \ldots, r_n)$. The final linear layer maps the refined representations to the BIO label probabilities:

$$\mathbf{p}_i = \text{softmax}(\mathbf{W} r_i + \mathbf{b})$$

where $\mathbf{W} \in \mathbb{R}^{k \times d}$ and $\mathbf{b} \in \mathbb{R}^k$ are learnable parameters. $k = 13$ is the number of BIO labels.

The model is trained to minimize the standard cross-entropy loss:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log p_{ij}$$

where y_{ij} is the true BIO label for token x_i and BIO label j, and p_{ij} is the predicted probability of token x_i having BIO label j.

4.3 Notes on Longformer and PII

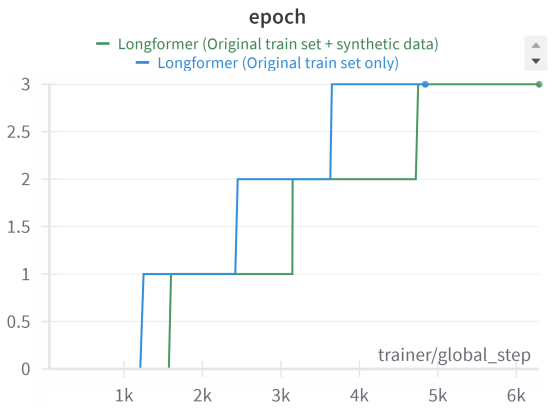
The Longformer's local attention mechanism is confined to small windows surrounding each token. By attending to the local context, the model remains responsive to the syntactic structure of the text—essential for accurately identifying PII entities such as student names, often contextually dependent on nearby words.

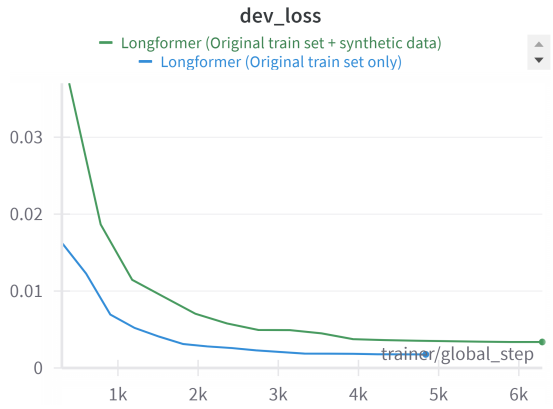
On the other hand, global attention allows select tokens to attend to the entire text, enabling the model to integrate broader narrative contexts. This is particularly useful in essays where references to PII can be subtly woven into the text or contextualized across different sections of the same essay. By maintaining an awareness of the entire essay, the Longformer can better understand the relevance and implications of specific tokens, enhancing its ability to discern PII from similarly structured non-PII text.

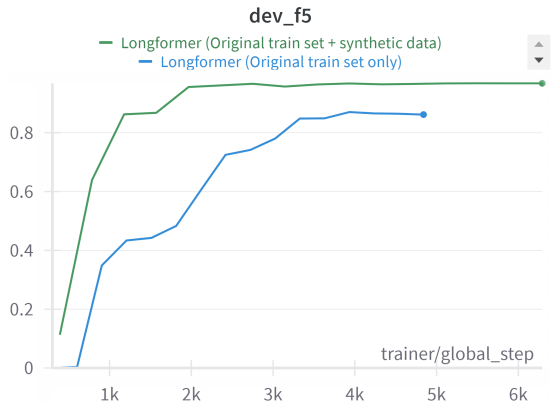
The combination of local and global attention mechanisms allows the Longformer to effectively navigate the dual challenges of length and complexity in student essays. By focusing on local relevance while simultaneously integrating global context, the model strikes a balance between depth of focus and breadth of understanding. This makes it particularly well-suited for the nuanced task of detecting PII in essays, where the context and narrative structure of the essays play a significant role in identifying sensitive information.

4.4 Training runs

Charts generated by Weights & Biases (Biewald, 2020).







5 Methods

5.1 Benchmark: Presidio

The benchmark is Presidio, a data protection and anonymization SDK developed by Microsoft. Presidio (Mendels et al., 2018) consists of two main modules: a recognition engine and an anonymization engine. The recognition engine uses a combination of techniques to detect PII entities: regular expressions, NER using spaCy, and predefined context words. (I'm only interested in the recognition engine for the benchmark.)

Presidio was configured with its default settings. As per one of the benchmarks used in Pilán et al. 2022, organization names are excluded to maintain a focus on *personal* information. Considerations for adapting Presidio to the test set for my experiments:

- Map Presidio's PII entity types to the corresponding types in the test set (for example, `PERSON` to `NAME_STUDENT`). See Table 3 for full map.
- Handling multi-token entities and assigning appropriate BIO labels (`B-` for the first token, `I-` for subsequent tokens within the same entity).
- Addressing potential overlaps between different entity types (for example, ensuring that a URL is not labeled as an `EMAIL_ADDRESS` if it is part of a larger `URL` entity).

Table 3: Map of Presidio entity types to full labels.

Presidio entity	→	Corresponding type
`PHONE_NUMBER`	→	`PHONE_NUM`
`PERSON`	→	`NAME_STUDENT`
`URL`	→	`URL_PERSONAL`
`EMAIL_ADDRESS`	→	`EMAIL`
`US_SSN`	→	`ID_NUM`

5.2 Inference on fine-tuned Longformer models

The maximum sequence length for inference is 4096.

If the probability of the `O` label is below a threshold of 0.85, the non-`O` BIO label with the highest probability is selected.

6 Metrics

To comprehensively evaluate the performance of Presidio and my original Longformer models in identifying PII in the test set, I used standard classifier metrics: precision, recall, and a weighted F-score.[3]

Precision, in the context of this project, measures the proportion of correctly predicted positive PII BIO labels among all the BIO labels predicted by the model. It reflects the model's ability to avoid false positives—a higher precision indicates the model is more likely to be correct when it predicts a token as PII. Precision is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall measures the proportion of correctly positive PII BIO labels among the true set of PII BIO labels. A high recall represents a low proportion of PII tokens left undetected by the model—indicating

[3] As discussed in CS224U: NLP methods and metrics

Table 4: Model names and abbreviations.

Model name	Checkpoint size	Abbreviated name
Longformer (Original train set only)	609.0MB	`Lf(O)`
Longformer (Original train set + synthetic data)	609.0MB	`Lf(O+S)`

the degree of privacy protection. Recall is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

To quantify the balance between privacy protection and data utility preservation, I used the F_β score. F_β is the standard weighted harmonic mean of precision and recall—β is the parameter that controls the relative importance of recall over precision. The F_β score is calculated as:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

I used $\beta = 5$. Setting β to 5 assigns five times more importance to recall over precision, reflecting the precedence of privacy protection over data preservation in the context of strict data anonymization requirements.[4]

I compute the precision, recall, and F_5 score for each PII type (see full list) as well as the micro-averaged scores across all entity types. The micro-averaged scores assess the model's overall performance, considering the dataset as a single large document.

7 Results

The top-line results are in Table 5.

Table 5: F_5 values for different models.

Model	F_5
Presidio	0.735
`Lf(O)`	0.831
`Lf(O+S)`	**0.936**

See Table 6 for granular results.

[4]The Kaggle leaderboard for the competition uses this metric, so in case I want to submit my original model to the competition, it will be scored according to it.

8 Analysis

8.1 Core analysis

The metrics in Table 5 show that the `Lf(O+S)` model achieves the highest overall F_5 score of 0.936, substantially outperforming both the Presidio benchmark ($F_5 = 0.735$) and the `Lf(O)` model ($F_5 = 0.831$).

This validates the primary hypothesis that the benchmark [Presidio] will be significantly outperformed, as measured by standard classifier metrics [F_5], by a custom model fine-tuned on the train set of a high-quality, human-annotated, domain-specific dataset [`Lf(O)`].

Looking at the category-level metrics in Table 6 provides further insight. The `Lf(O+S)` model achieves near-perfect F_5 scores above 0.95 for the most common categories of PII in original train set `NAME_STUDENT` and `URL_PERSONAL`. We clearly see F_5(`Lf(O+S)`) $>$ F_5(`Lf(O)`) $\gg$ F_5(Presidio) for these catagories. This supports the secondary hypothesis that the primary hypothesis holds principally because of *in-context* PII, which is highly domain-specific. Simply put, the more examples of this *in-context* PII category that are learned—either from the original test set or the synthetic data—the better the fine-tuned Longformer model performs.

Interestingly, Presidio achieves a near-perfect F_5 score of 0.987 for `EMAIL`. Presidio uses a regular expression to identify this PII type, which beats the pure machine learning approach.

8.2 Improvements and further research

- One of the most surprising and interesting results of the experiments was the utility of synthetic data for significantly bolstering the performance of the Longformer model, particularly for categories with very few examples in the original train set. For instance, the F_5 score for `PHONE_NUM` improved from 0.000 in `Lf(O)` to 1.000 in `Lf(O+S)`. Even for categories with relatively abundant examples, such as `NAME_STUDENT`, the F_5 score increased by 0.065 with the addition of synthetic

Table 6: Precision (P), Recall (R), and F_5 values for different categories and models.

PII type	Presidio			Lf(O)			Lf(O+S)		
	P	R	F_5	P	R	F_5	P	R	F_5
NAME_STUDENT	0.190	0.918	0.800	0.691	0.905	0.894	0.745	0.970	**0.959**
URL_PERSONAL	0.0444	1.000	0.547	0.333	1.000	0.929	0.480	1.000	**0.960**
EMAIL	0.750	1.000	**0.987**	0.000	0.000	0.000	0.667	0.667	0.667
PHONE_NUM	0.250	1.000	0.897	0.000	0.000	0.000	1.000	1.000	**1.000**
STREET_ADDRESS	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.909	**0.912**
ID_NUM	0.000	0.000	0.000	1.000	0.375	0.384	0.333	0.500	**0.491**
USERNAME	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

data. These findings suggest that the performance of the PII detection model could be further improved by conducting a more in-depth analysis of the train set structure and adopting a more systematic approach to generating fake PII sets and essay generation prompts. It would be interesting to explore the extent to which the original train set could be replaced by generated synthetic data without compromising the model's performance.

- To enhance the model's performance on categories with limited examples, such as ID_NUM and USERNAME, a fine-grained analysis of the existing examples in the train set should be conducted.

- For real-world PII detection pipelines, a hybrid approach that combines the strengths of rule-based mechanisms, such as those employed by Presidio, with the fine-tuned Longformer model could yield promising results. Rule-based techniques can efficiently capture well-defined patterns and formats, while the Longformer model excels at understanding the nuanced context and identifying PII that may not adhere to strict rules. By leveraging the complementary strengths of these two approaches, a more comprehensive and robust PII detection system could be developed.

- To strengthen the Presidio benchmark, future work could explore extending its default recognizers with custom ones tailored to specific contexts, as demonstrated by Aziz and Straiton 2023 in their work on identifying Australian-specific PII types. In the context of this project, custom recognizers could be developed to better capture PII patterns specific to the educational domain or the United States, such as student ID formats, school-related abbreviations, or state-specific information. By incorporating these domain-specific recognizers, the Presidio benchmark would become a more challenging and relevant baseline for evaluating the performance of the Longformer model.

Known project limitations

- The corpus comprises essays submitted in the context of a MOOC hosted by a university in the United States. We can assume most students are based in the United States. This means the models may not generalize well to identifying PII with local context outside the United States, such as identifying non-US phone number formats, addresses, or region-specific identification numbers.

- The performance of the models may be sensitive to variations in PII formats or representations. For example, if a student writes their phone number in an unconventional way ("five-five-five, one-two-three, four-five-six-seven"), the model might not recognize it as a phone number. Robustness to such variations is an important consideration for practical applications.

- The performance of the models on multilingual essays or essays containing non-English PII has not been evaluated.

Authorship statement

This final project report was produced using the *LaTeX* template c/o Christopher Potts.

This was an individual project for a Stanford Center for Professional Development course

(XCS224U) and there were no external contributors, reviewers or editors.

References

Ajmal Aziz and Rachael Straiton. 2023. Pii detection at scale on the lakehouse. YouTube. Channel: Databricks. Available at: https://www.youtube.com/watch?v=nTAKQuxZ9lI.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Yukun Chen, Thomas A. Lasko, Qiaozhu Mei, Joshua Charles Denny, and Hua Xu. 2015. A study of active learning methods for named entity recognition in clinical text. *Journal of biomedical informatics*, 58:11–18.

Daniele Faraglia and Other Contributors. 2023. Faker.

Rajitha Hathurusinghe, Isar Nejadgholi, and Miodrag Bolic. 2021. A privacy-preserving approach to extraction of personal information through automatic annotation and federated learning. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, pages 36–45, Online. Association for Computational Linguistics.

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Omri Mendels, Coby Peled, Nava Vaisman Levy, Tomer Rosenthal, Limor Lahiani, et al. 2018. Microsoft Presidio: Context aware, pluggable and customizable pii anonymization service for text and images.

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.

Thomas Vakili, Anastasios Lamproudis, Aron Henriksson, and Hercules Dalianis. 2022. Downstream task performance of BERT models pre-trained using automatically de-identified clinical data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4245–4252, Marseille, France. European Language Resources Association.

Niek van der Plas. 2022. Detecting pii in git commits. master thesis, Delft University of Technology, 07. TU Delft Electrical Engineering, Mathematics and Computer Science.