



Deidentifying Student Writing with Rules and Transformers

Langdon Holmes¹✉ , Scott A. Crossley¹ , Wesley Morris¹ ,
Harshvardhan Sikka² , and Anne Trumbore³

¹ Vanderbilt University, Nashville, USA

langdon.holmes@vanderbilt.edu

² Georgia Institute of Technology, Atlanta, USA

³ University of Virginia, Charlottesville, USA

Abstract. As education increasingly takes place in technologically mediated settings, it has become easier to collect student data that would be valuable to researchers. However, much of this data is not available due to concerns surrounding the protection of student privacy. Deidentification of student data is a partial solution to this problem, but student-generated text, a form of unstructured data, is a major challenge for deidentification strategies. In response to this problem, we develop and evaluate two approaches for the automatic detection of student names. We develop one system using a rule-based approach and one using a transformer-based approach that relies on finetuning a pretrained large language model. Our findings indicate that the transformer-based approach to student name detection shows more promise, especially when there is a high degree of variation between texts in a dataset.

Keywords: deidentification · anonymization · large language models · massively open online course · named entity recognition

1 Introduction

Educational artificial intelligence systems rely on large amounts of student data. However, as learning technologies make it easier to collect and utilize this data, there are growing concerns surrounding student privacy. Anonymization of student data is an important strategy for developing educational tools while protecting student privacy.

The first step in automated deidentification is to label personally identifiable information (PII). Rule-based approaches to labeling work by applying a set of rules or labeling functions to a text. For instance, Lison et al. developed a Python library to facilitate the creation of rule-based labeling systems, Skweak [3]. In an application, Skweak was used to achieve an F1 of 81% on per-token named entity labeling in a corpus of Wall Street journal articles, which demonstrates the potential for rule-based approaches to detect PII in unstructured text. In the educational domain, Bosch et al. used a rule-based approach that aggregated a

set of text-level features using a machine learning model. They achieved as high as 95% recall of student names in a university classroom’s discussion forum [1]. Deep-learning-based approaches to PII labeling rely on transformers, which are a neural network architecture that is widely used in natural language processing. Previous work has shown success in the medical domain with this approach, achieving PII recall as high as 99% on some medical datasets [5].

In the current study we build and assess a transformer-based student name labeling system alongside a rule-based system. We focus on student names because they are the most prevalent form of PII in student-generated text [2] and because they are challenging to detect. Our rule-based name labeling system works by aggregating a set of rules that detect student names. Our transformer-based name labeling systems were developed by finetuning a pretrained transformer model on domain-specific labeled training data. We evaluate these systems to assess their potential as part of an automated deidentification system for student-generated text.

2 Methods

Student writing samples were collected by Coursera from students enrolled in a publicly available, online course. Course completion required students to submit a written essay in which they reflected on how the course content could be applied to a problem with which they are familiar. Submissions were required to be in PDF format, and the files were retained on a third-party hosting platform. In total, 221,043 submission events were recorded. 29,142 of these PDF files were converted to plain text. We selected 5,797 of the documents in the corpus for hand-coding of PII. Each document was annotated by two undergraduate students during an internship at a research university. The annotators were instructed to label any student names, including names that could refer to a classmate, and any names associated with the student (such as a colleague or a family member). The authors of cited texts, the instructors of the course, and public figures were not considered student names. Each document was seen by at least one annotator. All documents that included any student name annotations were reviewed by the first author to ensure accuracy. Student names were included in 845 submissions, and there were 1,155 student name annotations in total.

This dataset was split into training, validation, and test partitions, which comprised 60%, 20%, and 20% of the data, respectively. The training set was used exclusively to finetune the models used in our transformer-based system. The validation set was used for validating the transformer-based system during finetuning and for developing the rule-based name annotation system. The test set was used to evaluate the performance of both our rule-based and transformer-based systems. We report results only on the testing set for both systems.

2.1 Rule-Based System

We divide the task of labeling student names into two parts. The first part of the task is to identify which spans correspond to person names. The second part of the task is to identify which person names belong to students or might otherwise constitute PII. The system is illustrated in Fig. 1 and discussed below.

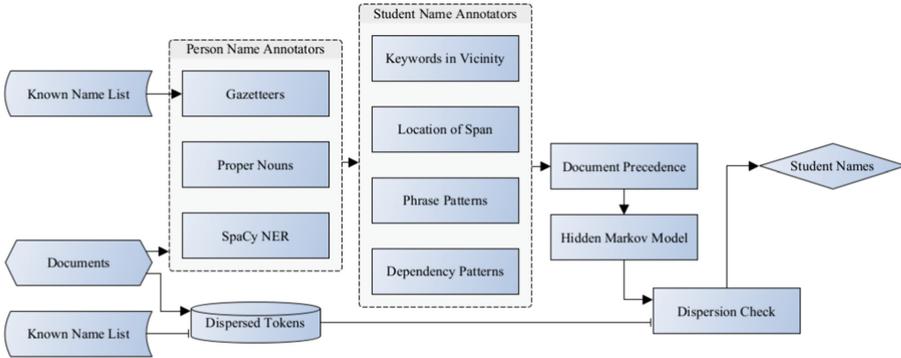


Fig. 1. A flowchart illustrating our rule-based student name labeling system.

We use three labeling functions to extract a set of person names from each document. These functions rely on part-of-speech tagging, a list of known names (called a gazetteer), and a general purpose NER model. The sole purpose of these functions is to supply a list of person names to the next group of labeling functions, which will determine which person names belong to students. We use a part-of-speech (POS) tagger to identify all proper nouns and label them as person names. We also search the document using a large list of known names and label any matches as person names [6]. We aggregate the output of these three labeling functions into a list of person name spans.

The next step is to identify which of these person names are student names. Vicinity annotators take a candidate span and search its context on the left and right hand side for a keyword token that might indicate it is a student name (e.g., the word “reflection” which often appears in the title). Location annotators consider the position of a span in the document. We created location annotators for spans that appear on a line by itself, for spans that appear in repeated lines, for spans that appear in the first lines of the document, and for the final span of the document. Phrase matcher annotators look for specific sequences of text, such as “Author: [proper noun].” We also developed syntactic patterns, such as “I am [proper noun].” Lastly, we use a precedence annotator to extend student name annotations to repeated mentions. If the person name “Brian” is labeled as a student name by any function, the student name label is extended to any repeated mentions of “Brian.”

In order to aggregate the output of all labeling functions into a single set of labels, we implement a hidden Markov model using Skweak. After the hidden

Markov model, we apply a post-processing step that checks for the dispersion of the labeled tokens throughout the full corpus of texts. Any token that appears in at least five documents is considered well dispersed, and therefore cannot uniquely and independently identify the author of the text. However, a common first name or other well-dispersed token may be used to identify a student in combination with other so-called quasi-identifiers. As a result, we reduce the list of named tokens using a set of known names extracted from Wikidata [7]. Any labeled spans which pass the dispersion check labeling function are the final student name labels.

2.2 Transformer-Based System

We developed two transformer-based models from Roberta, a pretrained large language model [4]. Each model used different weights for precision and recall during training. Sequence labeling tasks typically weight precision and recall equally. For the purposes of deidentification, we reason that recall should be weighted more highly because any false negatives could constitute a breach of student privacy. On the other hand, false positives are not as harmful. As a result, we trained the first model (Finetuned 1) with precision weighted at .25 and recall at .75. A second model (Finetuned 2) was trained with precision weighted at .10 and recall at .90.

We evaluate the performance of all systems in terms of recall, precision, and F1 score. The F1 score is the harmonic mean between precision and recall. Since the performance of the transformer-based system can only be evaluated on the testing set ($N = 1,160$), we use the same set to evaluate all our methods. There were 556 tokens that were part of student names in the labeled test set.

3 Results

We first evaluated the set of person name labeling functions on the held-out test set. Since documents were labeled for student names and not all person names, non-student person names are treated as false positives, resulting in lower-than-expected precision. Since all student names are person names, recall can be interpreted in a straightforward manner. The combined set of all person name labeling functions recalled 95% of student names with a precision of .03%. The best performing person name annotator was the general purpose NER, which recalled 81% of student names, with a precision of 33%. The full name gazetteer had a precision of 15%, which was higher than the other person name gazetteers, while also exhibiting a recall of 19%. The part of speech annotator, which labeled all proper nouns as persons, had the highest recall of any single labeling function. It recalled 95% of student names, which was the same as the combined results of all person name labeling functions. The part of speech annotator also had higher precision, .5%, than the combined system (.03%).

We then evaluated the student name labeling functions on the set of spans labeled as person names by the previous group of functions. The precision,

recall, and F1 score of each labeling function are reported in Table 1. The hidden Markov model, which aggregates the outputs of these functions, achieved a recall of .59 and a precision of .30. The dispersion filter, applied to the output of the hidden Markov model, resulted in an approximately unchanged recall of .59 and a precision of .41. The overall F1 score of this system was .48.

Table 1. Performance of student name labeling functions on the test set ($N = 1,160$).

| | Precision | Recall | F1 |
|-------------------------------|-----------|--------|-----|
| Dispersion filter | .41 | .59 | .48 |
| Hidden Markov model | .30 | .59 | .40 |
| Precedence annotator | .19 | .15 | .17 |
| Phrase Patterns | .35 | .17 | .23 |
| Dependency Patterns | .16 | .01 | .03 |
| Repeated line (header/footer) | .27 | .14 | .18 |
| Last span in document | .78 | .06 | .12 |
| Line by itself | .39 | .21 | .27 |
| Keyword Vicinity | .50 | .01 | .01 |

We tested two finetuned transformer-based models. The first model, Finetuned 1, was trained to weight recall at .75 and independently achieved a per-token recall of .64 and a precision of .90. The second model, Finetuned 2, was trained to weight recall at .90 and independently achieved a per-token recall of .84 and a precision of .68. Both of these models had approximately the same F1 score of .75, despite the different precision/recall trade off.

Finally, we created a third system that aggregates the labeling functions and both finetuned transformer-based models. This configuration resulted in a recall of .85 and a precision of .34. After applying the dispersion filter, recall was reduced to .83 and precision increased to .43. Overall, the combined system performs worse ($F1 = .56$) than the transformer-based systems alone.

4 Discussion

We evaluated the performance of rule-based and transformer-based approaches to the automatic detection of student names in student-generated text. We focused on student names as an important and challenging form of PII that has important implications in learning technologies. We developed two systems, specifically designed for our dataset, and evaluated their performance. Our results indicate that a rule-based system does not perform as well as a transformer-based system for student-generated text collected from a publicly available online course. Neither does the rule-based system contribute to the performance of the transformer-based system. We conclude that a rule-based

system is not well-suited to the complex task of student name labeling in highly variable student-generated text. Rule-based systems have proven to be effective for some text types, but they are brittle to formatting errors and stylistic variation that are to be expected in larger datasets collected from open courses. These challenges are particularly evident for the case of student names, which appear in a variety of linguistic contexts. An effective student name labeling system must also distinguish student names from other, non-student names such as referenced authors. Our transformer-based system was a more effective student name detector, suggesting that a generalizable deidentification system for student-generated text should adopt this approach for the detection of student names.

While the problem of automatic text deidentification is a challenging one, it does not appear intractable. There are few fields that would stand to benefit as much as learning analytics if an effective, automatic system were developed. Such a system, paired with ethical practices such as informed consent, would promote open science in while also protecting student privacy.

This material is based upon work supported by the National Science Foundation under Grant 2112532 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Acknowledgement. This material is based upon work supported by the National Science Foundation under Grant 2112532 Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Bosch, N., Crues, R.W., Shaik, N.: “Hello, [REDACTED]”: protecting student privacy in analyses of online discussion forums. In: Proceedings of The 13th International Conference on Educational Data Mining, p. 11 (2020)
2. Holmes, L., Crossley, S.A., Haynes, R., Kuehl, D., Trumbore, A., Gutu, G.: Deidentification of student writing in technologically mediated educational settings. In: Proceedings of the 7th Conference on Smart Learning Ecosystems and Regional Development (SLERD), Bucharest, Romania (2023)
3. Lison, P., Barnes, J., Hubin, A.: Skweak: weak supervision made easy for NLP. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pp. 337–346. Association for Computational Linguistics, Online, August 2021. <https://doi.org/10.18653/v1/2021.acl-demo.40>
4. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) [cs], July 2019
5. Murugadoss, K., et al.: Building a best-in-class automated de-identification tool for electronic health records through ensemble learning. *Patterns* **2**(6), 100255 (2021). <https://doi.org/10.1016/j.patter.2021.100255>
6. Remy, P.: Name dataset. GitHub (2021)
7. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun. ACM* **57**(10), 78–85 (2014). <https://doi.org/10.1145/2629489>