Title: Identifying Personally Identifiable Information (PII) in Student Writing
Team: ReLulu
Members: Christopher Stith, Katja Vassilev, Shirlyn Wang
Github: https://github.com/kdv97/pii

Description: The goal of our project was to identify Personally Identifiable Information (PII) in student essays. This helps automate the task of identifying and removing PII from student submissions to allow for distribution of materials and development of educational tools. This is part of a Kaggle Competition put out by Vanderbilt, in conjunction with The Learning Agency Lab, a nonprofit focusing on developing science and learning programs. Our goal was to assign labels to each word to minimize the F5 score, a metric that weighs recall 5 times as high as precision.

Data Overview: The data provided in the Kaggle Competition consisted of 6807 student samples, 5862 of which have no PII at all. The others have various types of PII such as names, emails, personal websites, ID numbers. The data was given to us both with the full text as well as a breakdown of the text into words and labels for each word, as well as if each word was followed by whitespace.

Main Approach: We chose to tackle this problem by fine tuning existing language models for the purpose of identifying the labels. This is an NLP task called Named Entity Recognition (NER). Two base language models that we thought were promising were RoBERTa (Robustly Optimized BERT model) and DeBERTa (Decoding Enhanced BERT with Disentangled Attention). Broadly, we fine-tuned these models by tokenizing our data with their tokenizer and feeding it into the model with the appropriate labels for each token. However, as the models only have a finite maximum token sequence length for training, we had to choose how to preprocess the data. We also decided to customize the fine-tuning a bit for improved performance.

Preprocessing Data: We decided there were two ways we wanted to preprocess the tokenized data so that we could feed it into the model:

- Truncating the data: A baseline model we found on kaggle simply took tokens up to the max training length, inspiring us to truncate the data. We figured that it was most likely that PII would occur at the beginning and end of the document, which was confirmed via analysis, so we took the approach of truncating out part of the middle. To avoid any PII that may occur in the middle, we analyzed specifically where we could truncate the data to minimize losing PII tokens.
- Partitioning the data: In order to not lose any text and maximize what was being fed to the model, we split each tokenized document which exceeded the model's maximum sequence length into several inputs.

Models: With these two main ways of preprocessing the data, we also experimented with downsampling some of the documents with no PII and altered the thresholds for making labeling decisions. In addition, we implemented a custom loss function in our fine-tuning to account for the fact that we are attempting to maximize F5, which weighs recall heavier. To do this, we

altered the weights of the existing cross-entropy loss function in the model to penalize incorrect non-PII labels.

Conclusion: Our initial baseline had an F5 score of 81.72. We were able to improve on this: our final model performance had an F5 score of 85.54. We were somewhat limited by our access to a GPU with large memory, which hindered how much testing we could do. A future direction that we also wish to explore is to generate more data for the model, as several of the PII labels were quite sparse. Given these limitations, we felt that we made a decent improvement on a baseline model as well as learned valuable skills in fine-tuning language models.

Acknowledgements: We would like to thank Valentin Werner on Kaggle, who publicly shared his Kaggle notebook, which served as our baseline model, and provided the general framework for our models.