Identifying PII in Student Writing

Team: ReLulu
Chris Stith, Katja Vassilev, Shirlyn Wang
University of Michigan

Explanation of Problem

- Personally Identifiable Information (PII) is a barrier to making educational materials open-access
- Identifying and removing PII allows course material to be shared online
- NLP task: Named-Entity-Recognition (NER)

Some Stats about Data

- Used data from a Kaggle competition hosted by Vanderbilt and the Learning Agency Lab (Nonprofit focusing on science learning and programs)
- Provided 6807 student samples, possibly containing student information such as
  - Names
  - Personal Websites
  - ID Numbers
  - Emails
  - Etc.
- Submissions were provided with the full text as well as
  - Separation of text into words
  - A label for each word
  - Other info about the text

Category	# Submissions
Student Name	891
Personal URL	72
ID Number	33
Email	24
Username	5
Phone Number	2
No PII	5862

Summary of Our Problem

- **Goal:** Categorize the words into the provided labels to maximize the F5 score
- **Metric**: F5, a metric which weights recall 5 times heavier than precision (ie. better to be generous in predicting PII labels)

$$F5 = \left(1 + 5^2\right) \frac{precision \cdot recall}{\left(5^2 \cdot precision\right) + recall}$$

Main Approach

- Finetune existing language models
  - Looked through a few Kaggle submissions to understand the basic structure of fine-tuning and then decided a few avenues we wanted to explore
    - Custom loss function
    - How to parse the data to put into the language model
    - Which language model to use: DeBERTa or RoBERTa
- Baseline Model:
  - DeBERTa
  - Cross-Entropy Loss
  - Maximum training length: 1024
  - Maximum inference length: 2048
  - Truncated Input Data

Baseline F5	81.72

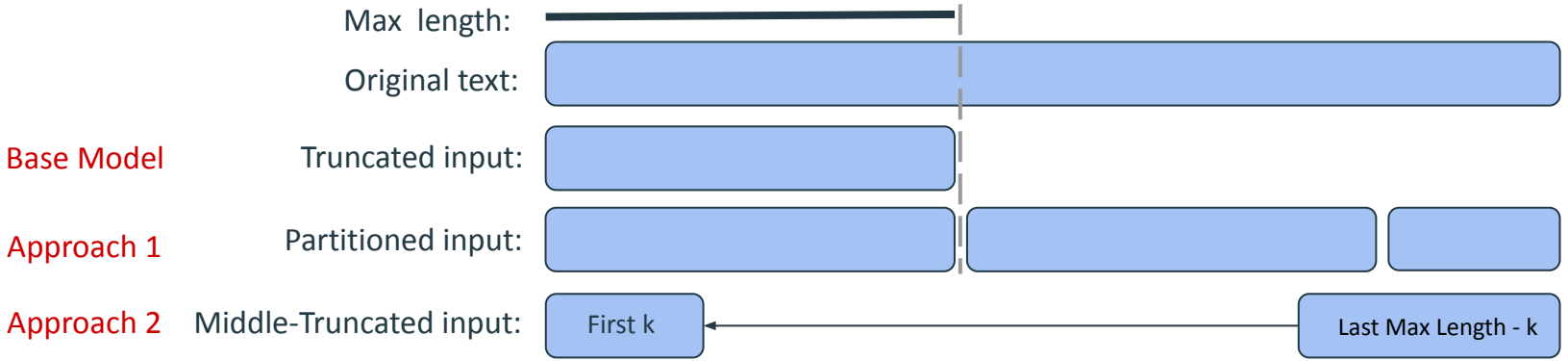
Custom Loss Function

- Cross-entropy loss function with customized label weights
  - Loss associated with misclassifying PII as non-PII is weighted ~20 times more than other misclassification

Model	F5
Baseline	81.72
Custom Loss	82.84

Processing data

- Each language model has a Max Length for tokenized input data (RoBERTa: 512, DeBERTa: 1024)
- Approach 1: Separate each text into several chunks, each of length less than Max Length
- Approach 2: Truncate data to the Max Length by cutting out the middle (less likely to have PII tokens)

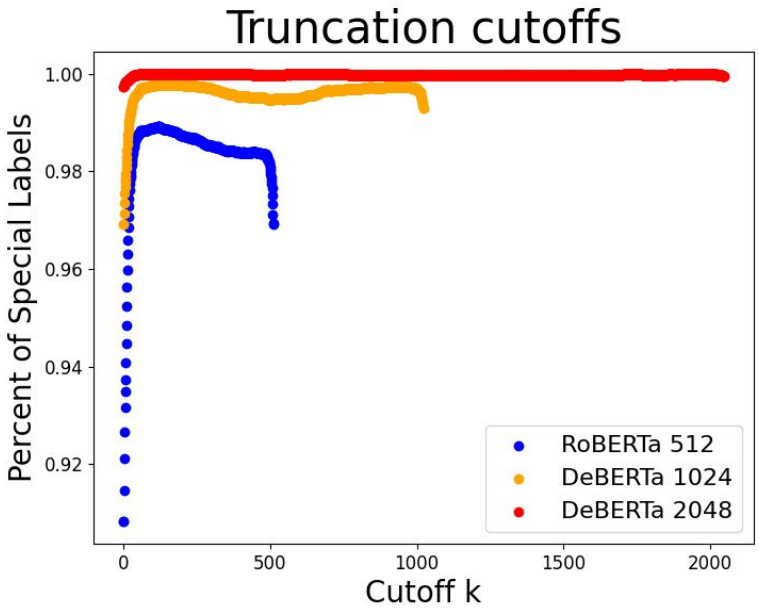


Approach 1: Partition the Data

- Improvements using both DeBERTa and RoBERTa
- Further implemented downsampling (in parentheses) for even better results on each language model
    - Partitioning leads to an larger percentage of inputs having no PII
    - PII was sparse to begin with

Model	F5
Baseline	81.72
Custom Loss	82.84
RoBERTa	83.89 (84.76)
DeBERTa	83.14 (85.70)

Approach 2: Truncate the data



- Only DeBERTa improves from baseline with alternate truncation as RoBERTa's max length for input size was far too restrictive

Model	F5
Baseline	81.72
Custom Loss	82.84
CL + T	84.97

Final Results

- We chose our best partition and truncation model (both DeBERTa)
- Further looked to improve results by altering the confidence thresholds used to decide what each token is.
- Finally, we trained on all the data
- ~4% Improvement

Model	F5
Partition	85.54
Truncation	85.37

Future Directions

- Limitations by access to GPU/Memory
  - We ran everything as Kaggle notebooks since they had the memory to fit the language models
  - 30 hr/week GPU (not enough time to fully test)
- Input more data to counteract sparsity of PII’s

Thank you!

- We would like to thank the following:
  - Erdos Institute for providing this Deep Learning Course, in particular Lindsay Warrenburg
  - Valentin Werner on Kaggle, for his baseline model and general framework