SLIDE 1
In this project I applied a modified version of ClinicalBERT on the Indiana chest X-ray dataset made available in the CAML HW assignment.

SLIDE 2
What is ClinicalBERT? ClinicalBERT is a pretrained version of BERT initially developed by Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Huang and his colleagues trained BERT on EHR data to enhance its capabilities in natural language tasks in the medical domain. Using their fine-tuned version of BERT, they were able to accurately predict 30-day hospital readmission from clinical discharge notes using the ADMISSIONS and NOTE_EVENTS tables in MIMIC-III.

SLIDE 3
The article that I intended to replicate for this project was not the original ClinicalBERT article, but a follow-up article entitled "ClinicalBERT: Using a Deep Learning Transformer Model to Predict Hospital Readmission, by Nwamake Imosagie.

Imosagie realized that rather than predicting readmission at the time of discharge, it would be more useful to predict readmission based on day-2 and day-3 notes *prior* to discharge, so that caregivers could intervene and prevent readmission before the patient had left the hospital. Imosagie achieved reasonably good results in her study, with evaluation accuracies of 63% for both days using the same MIMIC-III tables.

SLIDE 4
Although I initially prepared notebooks to preprocess data in the same manner as Imosagie, because I never ended up getting access to MIMIC-III data, at the suggestion of CS598's head TA Zhenbang Wu, I utilized the Indiana chest X-ray data made available in the CAML HW assignment. This meant going from a study that used 3 million records of data to one that contained less than 4000 rows of data. The data contained a subject ID, a clinical text description, and a Label comprising a multihot vector. The first field in the multihot vector represented whether the X-ray was normal (1) or abnormal (0), and so I used that one field to create binary values that could be used with Imosagie's code.

After performing NLP pre-processing on the data, I split the data using an 80-10-10 train/validation/test split. I then revised Imosagie's code base to make it work with the CAML data, reducing the LABEL to a binary: normal vs abnormal.

SLIDE 5

Training was performed using a batch size of 32, and the model was initially trained for 1 epoch, just as in Imasogie's original study. However, when the model failed to converge I tried running additional epochs, going as high as 8, with no effect. (Initial runs were on an old Macbook Pro, but when this proved too slow, I switched to Google Colab Pro). Unfortunately, as can be seen in the figure, the loss function continued to jump around, even when changing the learning rate.

SLIDE 6

As can be seen in these graphs, the area under the precision-recall curve (AUCPR) and the area under the receiver operating characteristic were both extremely low. AUCPR was 0.28, and AUROC was 0.26. I was actually stuck at these levels for over a weel, during which playing around with the hyperparameters made no difference whatsoever.

SLIDE 7

After staring at lists of threshold value predictions and y-labels, it dawned on me that the predicted and target values were wrong more often than they were correct. I further noticed that Imasagie had been predicting a "bad" thing as 1, whereas I was predicting "normal" (a good thing) as 1. And so it occurred to me that I would get much better results by simply reversing the polarity of the labels. And I reprocessed the CAML data accordingly.

SLIDE 8

And as it turned out, this made all the difference, as can be seen in the two figures shown here. AUPRC rose from 0.28 to 0.83, and AUROC went from 0.26 to 0.74. At this time, I also found that by increasing the size of the learning rate (to 5e-3), I got better results.

SLIDE 9

In this way, I was able to achieve evaluation accuracies on the order of 0.61, and an RP80 of 0.63. These values are on par with what Imosagie herself attained, and are quite satisfactory considering that my dataset was orders of magnitude smaller than

hers. Oddly, my result seems to suggest that ClinicalBERT does care about the polarity of the labels. This may be an artifact of the way ClinicalBERT was trained.