Preserving Privacy for Medical Data

Problem Statement

Deep Learning in healthcare can be used to streamline several administrative tasks as well as enabling physicians to make smart decisions based on the data. However, healthcare data is highly regulated and private making it inaccessible to perform deep learning tasks. In a medical data scenario, the Deep Learning model should be trained on the data without looking at it. In this project, we will explore ways (using PySyft and SyferText libraries) to enable secure and private Deep Learning to decouple sensitive data from the process of model training.

Goal

A classifier that correctly classifies the medical specialty based on the transcription text without looking at the dataset itself.

Dataset

Pre-processed Data: https://www.kaggle.com/tboyle10/medicaltranscriptions#mtsamples.csv

The original dataset is taken from mtsamples.com which contains 40 different medical specialties which in turn contain around 5000 transcribed medical reports under all specialties. The raw data has been pre-processed and converted into a CSV format for research purposes on Kaggle.

Note: These are only sample reports that are provided by various transcriptionists and users for reference purposes. In a real-world scenario, medical data is extremely hard to find due to privacy regulations.

Existing Work

Existing work [4] on this dataset involves using GloVe [5] and 1D CNN to classify the medical specialty. The benchmark results on this dataset are as follows.

Training Accuracy	Validation Accuracy	Recall	Precision	F1
0.87	0.89	0.88	0.98	0.93

These benchmark results are delivered using pre-trained models, however, training using a Bidirectional RNN approximately results in an accuracy of 68% [4].

Our Approach

As stated earlier our main goal for this project is to demonstrate private deep learning using OpenMined's PySyft and SyferText frameworks. There is a tradeoff between privacy and accuracy, i.e., models that leverage federated learning on encrypted text tend to produce lower accuracies than the models that run on a centralized server without any data encryption. The following are the steps required to solve our use case.

1. Using the PySyft framework create a bigger dataset out of all the client's smaller datasets.
2. Using SyferText to prepare and preprocess the text data on the client's machines without revealing it, and without moving any datasets to your machine.

Expected Outcome

Compare the accuracy achieved for the classifier with and without encryption. Our goal is to create a classifier that achieves an accuracy close to the original model.

References

1. https://github.com/OpenMined/SyferText
2. https://github.com/OpenMined/pysyft/
3. https://github.com/OpenMined/SyferText/blob/master/tutorials/usecases/UC01%20-%20Sentiment%20Classifier%20-%20Private%20Datasets%20-%20(Secure%20Training).ipynb
4. Marchawala, Alizar; Patel, Preetkumar; Paresh Thaker, Khushal; Gunjal, Hardik; nagrecha, Abhishek; Mohammed, Sabah (2020): Text Summarization and Classification of Clinical Discharge Summaries using Deep Learning. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.12059019.v1
5. https://nlp.stanford.edu/projects/glove/