CS598 DL4H in Spring 2022

Joseph Allen and Sonal Sharma
jaallen2@illinois.edu sonals3@illinois.edu

Group ID: 121, **Paper ID:** 215
Presentation link: TODO
Code link: https://github.com/josepha1/DLH_Final_Project
Difficulty level: Easy

1 Introduction

Our team is trying to replicate **Contextual Embeddings from Clinical Notes Improves Prediction of Sepsis** by Fatemeh Amrollahi, Supreeth P. Shashikumar, Fereshteh Razmi and Shamim Nemati for our Final Project.

Sepsis, a life-threatening organ dysfunction, is a clinical syndrome triggered by acute infection and affects over 1 million Americans every year. Early detection of sepsis and timely antibiotics administration is known to save lives. In this work the authors have designed a sepsis prediction algorithm based on data from electronic health records (EHR) using a deep learning approach. The findings indicate that incorporation of clinical text features via a pre-trained language representation model (ClinicalBert in this case) can improve early prediction of sepsis and reduce false alarms.

The proposed method replaces the traditional tf-idf features with contextual embedded representations learned using Clinical-BERT, a state-of-the-art model for word and document embedding and specifically trained on a corpus of Biomedical and clinical texts. The resulting document-level representations were then concatenated with physiological, laboratory, and demographic information to make predictions of sepsis using a long-short term memory (LSTM) recurrent neural network.

2 Scope of reproducibility

Contextual word embedding models such as ELMo and BERT have dramatically improved performance for many natural language processing (NLP) tasks. However, these models have been minimally explored in specialty corpora, such as clinical text; moreover, in the clinical domain, no publicly available pre-trained BERT models yet exist. ClinicalBert addresses this need by exploring and releasing BERT models for clinical text.

The paper claims that using a domain-specific model with LSTM for prediction yields performance improvements for sepsis prediction.

2.1 Addressed claims from the original paper

- Combining both structural clinical data with ClinicalBERT embeddings achieved the best AUC performance.
- The ClinicalBERT approach outperformed the TF-IDF model for extraction of features from clinical texts.
- ClinicalBERT generated more meaningful representations of clinical notes and enabled the model to predict onset of sepsis more accurately.

3 Methodology

We tried to re-implement the approach from the paper description as the author's code wasn't accessible to us. Publicly available MIMIC-III dataset was used to reproduce results similar to the paper.

3.1 Model descriptions

The paper uses ClinicalBERT and LSTM models for sepsis prediction and performance improvements.
ClinicalBert learns deep representations of clinical text using two unsupervised language modeling tasks: masked language modeling and next sentence prediction. A clinical note input to ClinicalBert is represented as a collection of tokens. In ClinicalBert, a token in a clinical note is computed as the sum of the token embedding, a learned segment embedding, and a position embedding. ClinicalBERT improves over BERT on the MIMIC-III corpus of clinical notes.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.

3.2 Data descriptions

Publicly available MIMIC-III (Medical information for intensive care) dataset of critically ill patients, which includes anonymized physiological and clinical data, as well as clinical notes from over 50,000 intensive care unit (ICU) admissions collected between 2001 and 2012 was used as part of the study. Data includes vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more.

MIMIC-III is a relational database consisting of 26 tables. Five tables are used to define and track patient stays: ADMISSIONS; PATIENTS; ICU STAYS; SERVICES; and TRANSFERS. Another five tables are dictionaries for cross-referencing codes against their respective definitions: D_CPT; D_ICD_DIAGNOSES; D_ICD_PROCEDURES; D_ITEMS; and D_LABITEMS.

Access to the PhysioNet database was requested and obtained following the steps mentioned in piazza.

3.3 Hyperparameters

The Hyperparameters used for training ClinicalBERT model:

- NUM_TRAIN_STEPS=100000
- NUM_WARMUP_STEPS=10000
- LR=5e-5
- train_batch_size=32
- max_predictions_per_seq=20
- save_checkpoints_steps=50000
- keep_checkpoint_max=15

The Hyperparameters used for training our final evaluation model(LSTM+Softmax):

- epochs = 25
- loss **=** 'mse'
- optimizer **=** 'adam'
- learning_rate =0.001
- batch_size=10

3.4 Implementation

Code Repository :
https://github.com/josepha1/DLH_Final_Project/blob/main/DLH-Project_final.ipynb

Our team had to implement our own code to replicate the results as the existing code from the authors' was not available. In order to properly analyze the clinical notes contained in the dataset, preprocessing helper functions needed to be implemented. These helper functions were responsible for removing any unnecessary punctuation or characters that would interfere with the prediction process. An additional purpose of these functions was to decompose the clinical notes into sentences and limit the size of the resulting sentences so that they could be fed to the ClinicalBERT model for processing. Our team then loaded the data tables that contained the clinical notes and the ICD9 diagnosis codes. These tables were combined so that the true labels for the diagnoses could be identified and later compared to the predicted results. The columns that were not necessary for our analysis were then dropped from the combined table and a final table was created for use with the ClinicalBERT model.

Due to computational limitations we were only able to conduct our analysis with a limited sample size of 500. We constructed a balanced dataset with 250 labels from each class, sepsis and non-sepsis. The sample dataset was later splitted into training and test sets with respective proportions of 80% and 20%.

For the TF-IDF model implementation, we vectorized the clinical notes from the dataset. Later, the features were transformed into arrays before splitting them into training and test sets. The final model (LSTM+Softmax) was trained and fitted to the training data in order to produce the predictions on the test set.

For the ClinicalBERT model implementation, the preprocessed sentences were first tokenized and fed into the ClinicalBERT model. Each processed sentence was then concatenated with the other processed sentences from each clinical note to create a feature vector per note. We split the dataset into training and test sets. The feature vectors from the training sample were then input into the Long-Short-Term-Memory (LSTM) neural network. The output of the neural network was finally input into a SoftMax activation layer to get the final prediction labels on the test sample.

We used Adam's optimizer and Mean Squared Error(MSE) to calculate loss on both the models.

3.5 Computational requirements

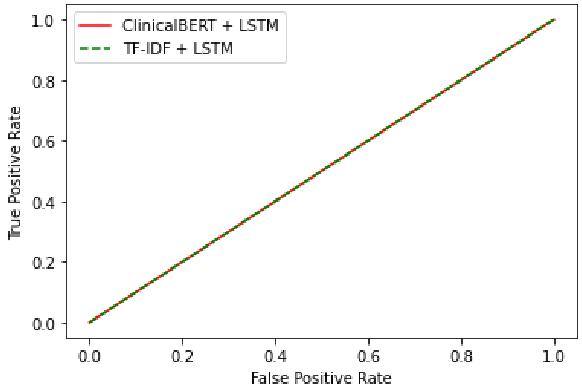
The code was implemented on a Dell XPS15 with 32 GB of RAM and an 11th Gen Intel® Core™ i7-11800H Processor operating at 2.30 GHz. The full dataset is approximately 4.2 GB, which is a moderately large dataset, took an average time of approximately 1 minute and 23 seconds to load. The average training time for the TF-IDF+LSTM model and ClinicalBERT+LSTM model was 1 minute and 43 seconds, and 1 minute and 32 seconds per epoch respectively. Preprocessing all of the data, inputting it into the ClinicalBERT model, as well as training the LSTM took significantly longer than we expected. The ClinicalBert model used by the authors' proved to be very computationally expensive on our hardware. We attempted to utilize the GPU offered by Google Colab, however, we were unsuccessful at processing larger amounts of data. We were also unable to find any reference on computational specifications used by the authors'.

4 Results

After inputting the preprocessed data into the ClinicalBERT and LSTM+Softmax network, our team obtained an accuracy of 0.425 whereas on the other hand the TF-IDF with LSTM+Softmax attained an accuracy of 0.5. Both the models attained AUC-ROC score of 0.5. The accuracy and AUC-ROC score indicates that the model is not accurately making predictions or distinguishing between classes very well. A possible explanation for the deviation of results from the original paper is that our team could not run the code on a large dataset due to computational limits.

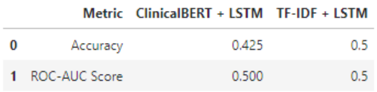
4.1 Result 1

Below is a graph of the AUC-ROC matrix for both the models.



4.2 Result 2

The below table describes the accuracy and ROC-AUC score of the ClinicalBERT vs TF-IDF model.



	Metric	ClinicalBERT + LSTM	TF-IDF + LSTM
0	Accuracy	0.425	0.5
1	ROC-AUC Score	0.500	0.5

4.3 Additional results not present in the original paper

In order to fetch better predictions on our clinical notes we additionally preprocessed data using two customized helper functions. It helped remove stop words and irrelevant punctuations. Also, split each clinical note into sentences for better efficiancy.

5 Discussion

We were not able to effectively reproduce the results of the original paper. As previously mentioned, our team believes that this was primarily due to computational limitations. The ClinicalBERT method implemented by the authors was computationally expensive, which limited the amount of data that we could process.

5.1 What was easy

The easy part of the implementation was preprocessing the data and utilizing the TF-IDF and ClinicalBERT methods. Executing this part of the process was relatively straightforward.

5.2 What was difficult

The difficult part of reproducing the methods of the original paper was structuring the sample of data so that it could be properly input into the LSTM neural network. Our lack of experience implementing this type of neural network resulted in a lot of time researching how to properly implement this process.

5.3 Recommendations for reproducibility

Our recommendations to the original authors would be to provide the computational specifications necessary to implement the methods they utilized in their research. Our recommendations to others who would like to reproduce the results obtained by the original authors would be to make sure that they have sufficient computational resources, as well as access to a GPU for increased computational Efficiency.

6 Communication with original authors

We attempted to contact the authors' of the original paper, but we were not succesful.

References

- *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8075484/*
- *https://github.com/EmilyAlsentzer/clinicalBERT*
- *https://github.com/nwams/ClinicalBERT-Deep-Learning--Predicting-Hospital-Readmission-Using-Transformer/blob/master/ClinicalBERT%20Deep%20Learning%20-%20Predicting%20Hospital%20Readmission.ipynb*
- *https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/*
- *https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76*
- *https://physionet.org/content/mimiciii/1.4/*
- *https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47*
- *https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-python-6c2b61b78558*