Reproducing, Validating, and Enhancing ClinicalBERT

Madhu Sivaraj and Anish Saha
{sivaraj4, saha9}@illinois.edu

Group ID: 67, Paper ID: 314
Presentation link: https://www.youtube.com/watch?v=KJ8UdyzkgoQ
Code link: https://www.github.com/madhusivaraj/clinicalBERT

1 Introduction

Clinical notes are often underutilized in the medical domain, given its high dimensionality, scarcity, and lack of structure. Unlike its structured, quantitative counterparts such as lab results, procedural codes, and medication history, clinical notes - with the help of deep learning models - have the potential to reveal high-quality, physician-assessed semantic relationships between medical concepts, which would otherwise involve a human perspective. Huang et al. (2020) devised ClinicalBERT, a flexible framework for learning deep representations of clinical notes, which can be useful for domain-specific predictive tasks [5]. Pre-trained on unstructured clinical text from MIMIC-III, ClinicalBERT leverages two unsupervised tasks, masked language modeling and next sentence prediction, followed by a problem-specific fine-tuning phase.

Our team hypothesized that ClinicalBERT would be an acceptable method for clinical language modeling and readmission prediction. Our goal was to reproduce, validate, and enhance ClinicalBERT, a model fine-tuned on a hospital readmission prediction task, using the ablations of data augmentation and a migration from the pytorch-pretrained-bert library to using the more sophisticated Transformers library. The results of our reproduced proposed, baseline, and ablation methods validate our hypothesis: ClinicalBERT outperforms all baseline models (that leverage Bag-of-Words, Bi-LSTM, and BERT) and our reproduced ClinicalBERT model improves AUROC and AUPRC by 22% and 19% and RP80 by 14% of the author's reported values.

2 Scope of reproducibility

In the original paper, Huang et al. improve upon previous clinical text processing methods by introducing ClinicalBERT. This approach learns deep representations of clinical notes and outperforms three baseline models—bag-of-words (BoW), bidirectional long-short term memory (BI-LSTM), and Bidirectional Encoder Representations from Transformers (BERT)—on a hospital readmission prediction task [1]. We were motivated to reproduce, validate and improve these results and test our hypothesis– that ClinicalBERT was a satisfactory method for clinical language modeling and readmission prediction in the healthcare domain.

2.1 Addressed claims from the original paper

- ClinicalBERT improves upon BERT on language modeling and next sentence prediction tasks, specifically in the context of the clinical notes data from MIMIC-III.
- ClinicalBERT outperforms competitive baseline models—BoW, Bi-LSTM, and BERT—on the basis of area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPRC), and recall at precision of 80% (RP80).
- Running ClinicalBERT with more data and fine-tuning should yield better performance.

2.2 Objectives for reproducing original paper

We postulated that models pre-trained on MIMIC-III clinical notes will have higher accuracy than baseline models trained with Word2Vec embeddings in readmission prediction tasks [7].

Our objective was to: 1) compare how ClinicalBERT and BERT perform in language modeling and next sentence prediction tasks; 2) reproduce ClinicalBERT and three baseline models (BoW, BI-LSTM, BERT) and validate if ClinicalBERT models have the highest AUROC, AUPRC and RP80 values; and 3) attempt to improve the accuracy and runtime of ClinicalBERT, even marginally, via a) data augmentation to enhance size and quality of training datasets and b) migration from pytorch-pretrained-bert to Transformers [6].

3 Methodology

To reproduce, validate and enhance Huang et al. (2020), we opted for a hybrid approach, leveraging open-source code provided by the original authors [4] and writing our own implementation for baselines, ClinicalBERT enhancements, and ablations.

3.1 Model descriptions

We reproduced all four models from the original paper, including the three baseline methods — Bag-of-words, BI-LSTM with Word2Vec, and BERT — and the proposed method, ClinicalBERT.

3.1.1 Baseline Method 1: Bag of Words

The first baseline model, bag-of-words, naively uses CountVectorizer to tokenize, build a vocabulary, and then encode the notes. CountVectorizer embeddings are used as input for a logistic regression model using L2 (ridge) regularization [9].

The learning objective is a logistic sigmoid function, optimized with limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) and L2 norm. The trained model has 15 parameters.

3.1.2 Baseline Method 2: BI-LSTM

The second baseline method, a bi-directional long short-term neural network, is a sequence processing model consisting of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. The BI-LSTM approach takes the Word2Vec model as its input word embedding, and the hidden state is fed into a global max pooling operation, and then a fully connected Dense layer with a rectified linear activation function. Dropout is applied, and followed by a Dense layer with 1 output unit and sigmoid activation function.

The learning objective is a binary classification function. Cross-entropy is applied and optimized with an Adam adaptive learning rate. The trained model has 484,902 parameters.

3.1.3 Baseline Method 3: BERT

The third baseline method, BERT, follows a multi-layer bidirectional Transformer encoder architecture [1]. It uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder to learn deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. $\text{BERT}_{\text{base}}$ takes CLS, a collection of classification tokens, as input. The inputs are passed to the 12 layers in the encoder stack. Each layer applies self-attention and passes the result through feedforward networks before moving it on to the following encoder. The model outputs a vector of hidden size; the classifier output can be derived by mapping to CLS token.

The learning objective is masked language model, a semi-supervised pretraining approach for models to predict a random sample of input tokens that have been replaced by a masked placeholder in a multi-class setting over the entire vocabulary [8]. The trained model has 109,483,009 parameters.

3.1.4 Proposed Method: ClinicalBERT

ClinicalBERT is a modified BERT model which applies bidirectional encoder representations to learn embeddings from clinical notes. The model takes CLS as input, and predominantly follows the architecture of BERT, with the exception of an added self-attention mechanism. Its self-attention mechanism is computed on an input sequence using associated clinical text embeddings. Each input embedding is multiplied by learned sets of weights to construct queries, keys, and values; for every query, a weighted combination of values calculated by the query and key are outputted. ClinicalBERT leverages two unsupervised pre-training tasks, masked language modeling and next sentence prediction, followed by a problem-specific fine-tuning phase.

The learning objective function is the sum of the log-likelihood of the predicted masked tokens and the log-likelihood of the binary variable indicating whether two sentences are consecutive. The trained model has 109,483,009 parameters.

3.2 Data descriptions

ClinicalBERT relies on Medical Information Mart for Intensive Care III (MIMIC-III), a database of deidentified electronic health information associated with 40,000 patients who stayed in critical care units of Beth Israel Deaconess Medical Center between 2001 and 2012 [2]. MIMIC-III includes unstructured data, including documented clinical notes by care providers and discharge summaries.

We obtained MIMIC-III (6.2GB) after gaining credentialed access to eICU Collaborative Research Database and PhysioNet [7] [2]. For the prediction task we were solving for, we merged two files, ADMISSIONS.csv and NOTEEVENTS.csv, to obtain clinical notes and discharge summaries for patients. After preprocessing, we had a dataset of 32,348 rows of clinical text data from discharge summaries and 59,011 rows from early notes. Both datasets were split using a relatively conventional

80-10-10 training-validation-test split. Each row had a binary target label with 1.0 or a 0.0, corresponding to whether or not the patient was readmitted, and both classes were nearly perfectly balanced 50-50 throughout the entire dataset.

3.3 Hyperparameters

For the BoW baseline model, the hyperparameters we set were: LBFGS as the solver for optimization; L2 as the regularization penalty norm; and 0.0001 as the C parameter which controls penalty strength. For the BI-LSTM baseline model, we set batch size to 64; loss function to binary cross-entropy; optimizer to the Adam algorithm; and dropout to 0.1. The hyperparameters for baselines were disclosed in a discussion we had with the author, but we extensively experimented with various hyperparameter configurations to optimize those results.

For BERT, ClinicalBERT and all ablations, we set learning rate to 0.00005; training batch size to 128; number of training epochs to 30, max sequence length to 512; and dropout rate to 0.1. However, for data augmentation ablations, we modified training batch size to 2 for computational reasons.

We set the hyperparameters by referencing the values specified in the appendix of the original paper and the existing code, but we used trial and error to choose the optimal training batch sizes for data augmentation ablation experiments.

3.4 Implementation

To reproduce the proposed ClinicalBERT method, we leveraged the publicly available code base[1] provided by the paper's authors [4]. We wrote our own code to implement the baseline models, enhance ClinicalBERT, and add ablations to the dataset, all of which can be found in our GitHub repository[2].

3.5 Computational requirements

We standardized our hardware requirements across all experiments for baselines, proposed model and ablations, running locally on a 2015 MacBook Pro with 8GB RAM and 2.7GHz dual-core processor.

The BoW model had a runtime of 0h5m, while the BI-LSTM model had a runtime of 1h31m for each experiment. Epochs are not applicable to the logistic regression model used for BoW, but BI-LSTM had 3 training epochs and the runtime per epoch of 0h30m. We ran 12 trials for both models; the results were within a margin of 0.02.

The BERT and ClinicalBERT models were less trivial. BERT had an total runtime of 4h35m and ClinicalBERT had an total runtime of 4h43m for each experiment. For all experiments conducted on both models, there were 3 training epochs. The runtime per epoch was 1h32m with BERT, and 1h34m with ClinicalBERT. We ran 12 trials for both, and the results were within a margin of 0.05.

The two ablations—data augmentation and Transformers migration—had a runtime of 5h12m and 1h20m respectively, for each experiment. While data augmentation did not take very long, training the models with the new data took a long time, about 4h40m for all 3 epochs, corresponding to an average runtime per epoch of 1h7m per epoch. For all experiments conducted on the modified ClinicalBERT model with Transformers, the number of training epochs was 3, so the average runtime per epoch was 0h27m. We ran 3 trials, and the results were within a margin of 0.01 each time.

4 Results

We were successful in reproducing Huang et al.'s baseline and proposed methods — fine-tuned on a readmission prediction task [5]. Our work supports all claims stated in 2.1, as our results validate that ClinicalBERT outperforms all baseline methods across AUROC, AUPRC and RP80 metrics.[3]

Table 1: Overall Results using discharge summaries

Proposed Method	AUROC	AUPRC	RP80
ClinicalBERT	0.748	0.723	0.276
Baseline Method	AUROC	AUPRC	RP80
Bag-of-words	0.697	0.660	0.210
BI-LSTM	0.702	0.690	0.115
BERT	0.501	0.510	0.004
Ablations	AUROC	AUPRC	RP80
Data Augmentation	0.795	0.760	0.672
Transformers	0.745	0.720	0.207

4.1 Result 1: ClinicalBERT improves over BERT on pretraining tasks with MIMIC

ClinicalBERT and BERT models generate CSV files illustrating its performance in language modeling and next sentence prediction tasks. A discussion with the original paper's author, Kexin Huang,

[1] http://github.com/kexinhuang12345/clinicalBERT
[2] http://github.com/madhusivaraj/clinicalBERT

[3] All metrics from our experiments are reported as the mean of 3 independent runs.

Table 2: Overall Results using early clinical notes

Proposed Method	AUROC	AUPRC	RP80
ClinicalBERT	0.823	0.810	0.597
Baseline Method	AUROC	AUPRC	RP80
Bag-of-words	0.675	0.660	0.058
BI-LSTM	0.723	0.700	0.238
BERT	0.501	0.510	0.004
Ablations	AUROC	AUPRC	RP80
Data Augmentation	0.823	0.810	0.597
Transformers	0.758	0.740	0.380

revealed these files could be used to compare predictions from physician-assessed semantic relationships to the model predictions for these unsupervised language modeling tasks. We used these files to calculate the accuracy of our model predictions for both tasks when trained on MIMIC-III.

Table 3: Model accuracy for language modeling tasks on discharge summaries and early clinical notes.

Model	Discharge Notes	Early Notes
ClinicalBERT	0.596	0.740
BERT	0.466	0.519

Our results reported in Table 3 enforces Huang et al.'s claim that ClinicalBERT improves upon BERT on MIMIC-III and our hypothesis that ClinicalBERT is better than BERT for clinical language modeling and readmission prediction tasks.

4.2 Result 2: ClinicalBERT outperforms the three baselines (BoW, BI-LSTM, BERT)

Comparing the results of our reproduced proposed and baseline models with values reported in Huang et al. (2020) for 30-day readmission prediction using discharge and early clinical notes, ClinicalBERT has the highest AUROC, AUPRC and RP80 scores in all contexts [5]. Reported metrics in Table 4b and 5b are the mean of 3 independent runs.

Using discharge summaries, we improved AUROC and AUPRC by 4.7% and 3.1% of the reported values for ClinicalBERT, and implemented the baseline methods for which our results either outperformed or were, on average, within 5% of the reported values. When using early clinical notes, we observe that our ClinicalBERT improved AUROC and AUPRC by 22% and 19% of the author's reported values. The results of baseline methods also outperform their respective reported value, with the exception of BERT which remains within a 20% margin. These insights support Huang et al.'s claim that ClinicalBERT outperforms all baselines.

Table 4: 30-day readmission using discharge summaries

Model	AUROC	AUPRC	RP80
ClinicalBERT	0.714	0.701	0.242
Bag-of-words	0.684	0.674	0.217
BI-LSTM	0.694	0.686	0.223
BERT	0.692	0.678	0.172

(a) Results obtained from original ClinicalBERT paper

Model	AUROC	AUPRC	RP80
ClinicalBERT	0.748	0.723	0.276
Bag-of-words	0.675	0.660	0.210
BI-LSTM	0.702	0.690	0.115
BERT	0.501	0.510	0.004

(b) Results obtained from our reproduction of models

Table 5: 30-day readmission using early clinical notes

Model	AUROC	AUPRC	RP80
ClinicalBERT	0.672	0.677	0.170
Bag-of-words	0.654	0.657	0.122
BI-LSTM	0.656	0.668	0.150
BERT	0.661	0.668	0.167

(a) Results obtained from original ClinicalBERT paper

Model	AUROC	AUPRC	RP80
ClinicalBERT	0.823	0.810	0.597
Bag-of-words	0.675	0.660	0.058
BI-LSTM	0.702	0.700	0.238
BERT	0.501	0.510	0.004

(b) Results obtained from our reproduction of models

When comparing evaluation accuracies in Table 6, ClinicalBERT dominates on this metric as well with a respective 65% and 74% accuracy on discharge and early notes. BoW and BI-LSTM follow at a distance, with 62% and 61% accuracy. This further upholds our hypothesis as ClinicalBERT is an acceptable methodology compared to the baselines.

4.3 Result 3: Running model with more data and fine-tuning yields better performance.

Huang et al. (2020) suggests that ClinicalBERT be improved by running the model on a larger collec-

Table 6: Evaluation accuracy of four reproduced models

Rank	Model	Discharge	Early Notes
1	ClinicalBERT	0.647	0.740
3	Bag-of-words	0.611	0.605
2	BI-LSTM	0.616	0.606
4	BERT	0.466	0.466

tion of notes for better performance. We added ablations and ran additional experiments not present in the original paper, including a) data augmentation; and b) migration to Transformers.

4.3.1 Ablation 1: Data augmentation

To enhance the size and quality of training datasets, we tried two different methodologies: i) increase the training dataset size by increasing the time window in which notes could be counted as "early" and ii) by introducing synonym replacement and random swapping to create new rows of data.

We augmented the training dataset of clinical notes from both the discharge and early readmission categories by modifying text, because in most cases, a larger training dataset helps to improve performance. In the context of other deep learning tasks, such as image recognition, similar data augmentation methodologies (adding new training data using existing training samples via transformations and random noise) have helped significantly improve performance, so this was our motivation. We performed NLP-based data augmentation by performing random synonym replacement and word swapping. This not only helps to synthesize a larger dataset, but also prevents overfitting and promotes generalization [3]. Using data augmentation to increase the training dataset size to 52,490 training samples and 95,586 training samples for the discharge and early readmission tasks, respectively, we were successfully able to create a larger dataset on 5 and 7 days to run additional experiments on.

Table 7: 30-day readmission using augmented data

Clinical Notes	AUROC	AUPRC	RP80
Discharge (aug.)	0.795	0.760	0.276
Early (aug.)	0.823	0.810	0.597
Early ($\leq$5days)	0.796	0.800	0.521
Early ($\leq$7days)	0.763	0.780	0.536

Contrary to our predictions, none of the data augmentation methodologies helped to improve performance at all. The method leveraging random replacement and synonym swap attempted to build on top of the existing ClinicalBERT models by feeding in more data (with the same meaning), but the resultant model was extremely similar, so performance remained constant. With the case of redefined early notes categorizations, new meaningful data was added, but the model was not successfully able to learn from that data to be able to generalize word associations and improve performance. In order to truly validate the claim that a larger dataset and fine-tuning would improve performance, new clinical notes datasets and increased computational resources would need to be explored.

4.3.2 Ablation 2: Transformers

Transformers (formerly pytorch-pretrained-bert) is a library of over 32 state-of-the-art pre-trained models for natural language processing [6]. We updated our ClinicalBERT model with Transformers, reimplementing the forward method to always output a tuple with encoded layers and pooled output. The two optimizers—BertAdam and OpenAIAdam—have been replaced by a single AdamW optimizer, so we replaced BertAdam with AdamW, following the same decay schedule.

Table 8: Performance of ClinicalBERT, migrated to using Transformers over pytorch-pretrained-bert

Clinical Notes	AUROC	AUPRC	RP80
Discharge	0.745	0.720	0.207
Early	0.758	0.740	0.380

We ran 3 independent trials, with a batch size of 3, learning rate of 0.00005, hidden size of 768, and dropout rate of 0.1. Table 7 reveals that while our reproduced ClinicalBERT had a combined runtime of 9h26m for both experiments, the ablations with adding Transformers resulted in a 84% decrease to 1h28m. Table 1 and 2 also reveal the AUROC and AUPRC values of reproduced and modified ClinicalBERT models, on discharge and early clinical notes, are within a 1% margin of each other.

Given the lower compute time and higher AUROC and AUPRC for both tasks, these findings reinforce Huang et al.'s speculation and our hypothesis that finetuning may yield better performance.

5 Discussion

Overall, our experimental results show that the original authors' work was indeed reproducible;

we were largely able to validate their main claim that ClinicalBERT is significantly more effective for modeling clinical language, and consequently, for the readmission prediction task. Higher AUROC, AUPRC and RP80 metrics observed when ClinicalBERT trained on early clinical notes versus discharge summaries indicates the model's efficacy of predicting readmission with unstructured data.

We were unable to validate if larger amounts of data would improve performance. Given access to similar datasets and better computational resources, we might have been able to validate the authors' suspicion; however, we faced difficulties as the preprocessing-training pipeline was computationally expensive, with each experiment taking 12h to complete. However, we validated that finetuning could yield better performance. Migrating from pytorch-pretrained-bert to Transformers helped reduce runtime by 84% to 1h30m per experiment. Thus, we find ClinicalBERT to be effective for clinical language modeling and readmission prediction.

5.1 What was easy

Reproducing the results for the proposed method, ClinicalBERT, was not too challenging, as we were able to leverage and re-purpose the author's code. Although we had to make modifications in the preprocessing pipeline and update deprecated code in some models, reproducing and experimenting with the ClinicalBERT results was simple. Running experiments for the Bag-of-words baseline was not complicated because, compared to other methods, the computational resources needed was much lower ($\leq$1 hour on a local machine). We found it easy to understand the paper and its technical details, as things were generally well-documented.

5.2 What was difficult

Reproducing and enhancing this paper was quite the challenge. To start, reproducing the BoW and BI-LSTM baselines were difficult because no code was provided by the authors to reference. The same could be said for implementing our ablations (migration to Transformers and data augmentation pipeline). These pipelines had to be implemented from scratch with minimal resources available. Despite the scarcity of documentation, we relied on course material, online tutorials, and much trial-and-error to eventually achieve success. We also struggled to reproduce the baseline results reported in the paper (Bag-of-Words, Bi-LSTM, and BERT). However, after discussing model architecture, libraries, and hyperparameter settings with the author, we were able to reproduce those results. The BERT baseline provided was broken so we had to debug and remediate the issue. Performance was lower than expected, and it was tough to improve.

The most difficult part of this project was running experiments. Completing each experiment took on average 6h, so experimenting with hyperparameters was an inefficient, albeit necessary, use of time. In addition, training new models was not possible, as the code and the commands provided were broken. All 30+ attempts resulted in errors involving sephamore memory leaks. We tried communicating with the original author and extensively debugging the issue on our own, but we were unsuccessful. As such, we could only recreate and improve upon existing models (early, discharge, and pretrained BERT, ClinicalBERT); we tried to run those models with our augmented datasets, but, as expected, it did not work. Despite its challenges, this project was truly an enlightening experience.

5.3 Recommendations for reproducibility

Our suggestions are: (i) Do not run experiments locally. The dataset is large, with 2M clinical notes, and training time is long, averaging 6h per experiment. (ii) Consider ablations involving domain-specific embeddings. (iii) Keep health data privacy and security concerns in mind when using MIMIC-III, as per agreements with MIT. (iii) Train new models with another dataset to compare and validate performance. (iv) Run multiple experiments and account for standard deviation between results.

6 Communication with original authors

We corresponded with the original author, Kexin Huang, several times over the course of 6 weeks to gain his insight and perspective on our decisions and process. We discussed the following: (i) general pipeline; (ii) specific hyperparameter tuning information necessary to reproduce baseline results; (iii) ideas for ablations (he agreed that data augmentation and migrating from pytorch-pretrained-bert to Transformers were a good call, even if the performance diminished by 1%. They were necessary modifications to be made); and (iv) if FastText and TD-IDF would be worthwhile ablations given their low Pearson correlation coefficient in his research.

We shared our results and insights with Kexin for his input, which was positive. Overall, our communication with Kexin was productive and enriching.

References

[1] Jacob Devlin. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. URL: https://arxiv.org/pdf/1810.04805.pdf.

[2] *eICU Collaborative Research Database*. URL: https://eicu-crd.mit.edu/gettingstarted/access/.

[3] Pema Garg. *NLP data augmentation*. Feb. 2022. URL: https://pemagrg.medium.com/nlp-data-augmentation-a346479b295f.

[4] Kexin Huang. *ClinicalBERT: Modeling clinical notes and predicting hospital readmission*. URL: https://github.com/kexinhuang12345/clinicalBERT.

[5] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. *ClinicalBERT: Modeling clinical notes and predicting hospital readmission*. Nov. 2020. URL: https://arxiv.org/abs/1904.05342.

[6] HuggingFace. *Transformers*. URL: https://huggingface.co/docs/transformers/index.

[7] Alistair Johnson, Tom Pollard, and Roger Mark. *MIMIC-III Clinical Database*. Apr. 2019. URL: https://physionet.org/content/mimiciii/1.4/.

[8] *Open sourcing Bert: State-of-the-art pre-training for Natural Language Processing*. Nov. 2018. URL: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

[9] Ayush Pant. *Introduction to logistic regression*. Jan. 2019. URL: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148.