Reproducibility Project for CS598 DLH4H Spring 2023

Birhanu Digafe, May 2023
Group ID:29
Paper ID:16
Presentation link:https://youtu.be/egP30rpeSjE
Code link: http://github.org/bdigafe/cs598_dlh_final

1 Introduction

Deep learning, a subset of machine learning, is widely used in many healthcare applications, including predicting the onset of health failure [1]. Despite the availability of large amounts of electronic health records (EHR), they are still limited for predicting critical illnesses, especially those where early intervention can have significant positive effects. Previous techniques [8] relied on heavy feature engineering, which required significant expertise and effort. In contrast, deep learning models use raw patient data with minimal preprocessing and achieve better results [8].

This study aims to replicate the results of the BEHRT study, which was conducted on 1.6 million patients from the Clinical Practice Research Datalink (CPRD) [4] database in the UK. Due to data access restrictions, we used high-quality synthetic data. BEHRT used the BERT (Bi-directional Encoder Representations from Transformers) [3] architecture developed by Google for language models.

This study followed BEHRT's training approach, using BERT to encode medical conditions and create models that forecast the patient's future diagnosis of 296 conditions based on past EHR data. We believe with more quality data the approach can help for individualized medical predictions and prevention.

2 Scope of reproducibility

2.1 Alternative data source

To download the original source data, researchers are required to submit a request form to https://cprd.com/data-access, and access is subject to approval from the Independent Scientific Advisory Committee (ISAC). In short, the data was not readily available as a public data set. Therefore, we used synthetic EHR data from [5], which allowed us to download after registration. The data follows OMOP format, and it is smaller than the original study.

2.2 Reproducibility results

We were able to run all the tasks in the BEHRT study, i.e., embedding of 296 condition codes, MLM, next visit prediction, next 6 months, and next 12 months prediction tasks. Our MLM precision score was slightly lower compared to BEHRT. The results of the next disease predictions were much more closer. Due to resource constraints, we trained our model with 50 epochs while BEHRT was trained with 100 epochs. Still, our model outperformed the other compared approaches, i.e., Deepr [2] and RETAIN [2].

We ran t-SNE (t-distributed stochastic neighbor encoding) and created a 2D map of condition codes. Since our data set is not the same, we highlighted the top 20 closely matching conditions instead of those described in the paper. The results, however, is consistent with paper, i.e., the model identified closely matching conditions.

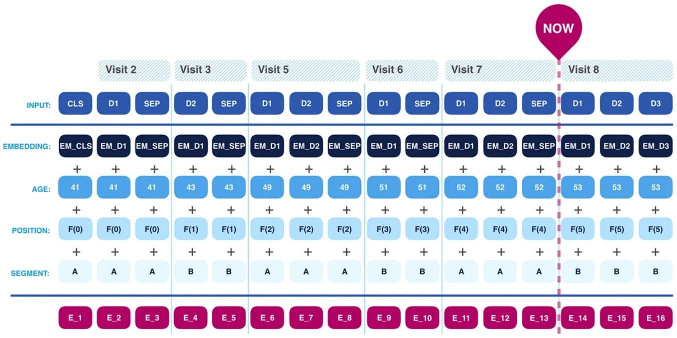
The paper didn't provide the code for hyper-parameter optimization. So, we tried using Bayesian Optimization, as described in the paper, using the Optuna [6] framework. However, we could not complete this task due to resource constraints on Google Colab. Instead, we used the same model parameters as basis, but also experimented with different parameters.

3 Methodology

3.1 Model description

The system uses BERT embeddings to generate a vector representation for each condition code based on its occurrence in the patient's medical history. This involves a multi-head self-attention layer, layer normalization, and a dense neural network layer with a non-linear activation function. Our model parameters are similar to BEHRT (12 attention heads, 6 hidden layers). The intermediate layer size is 51 and a total of approximately 4.5 million parameters.

The transformed input, i.e., the summation of all the embeddings, is fed to the BERT transformer-based architecture. First, the input is pre-trained using the MLM task where 86.5% of the condition codes remain unchanged, 12% are replaced with a mask, and the remaining 1.5% are replaced with random noisy codes. The model learns the network parameters, including the embedding representation of the disease codes and the masked disease codes. The precision score [7] of the pre-training MLM task is calculated for all patients and labels. The loss function is used to optimize the model parameters.



	Visit 2			Visit 3		Visit 5			Visit 6		Visit 7			Visit 8		
INPUT:	CLS	D1	SEP	D2	SEP	D1	D2	SEP	D1	SEP	D1	D2	SEP	D1	D2	D3
EMBEDDING:	EM_CLS	EM_D1	EM_SEP	EM_D1	EM_SEP	EM_D1	EM_D2	EM_SEP	EM_D1	EM_SEP	EM_D1	EM_D2	EM_SEP	EM_D1	EM_D2	EM_D3
	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
AGE:	41	41	41	43	43	49	49	49	51	51	52	52	52	53	53	53
	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
POSITION:	F(0)	F(0)	F(0)	F(1)	F(1)	F(2)	F(2)	F(2)	F(3)	F(3)	F(4)	F(4)	F(4)	F(5)	F(5)	F(5)
	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SEGMENT:	A	A	A	B	B	A	A	A	B	B	A	A	A	B	B	B
	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_10	E_11	E_12	E_13	E_14	E_15	E_16

Figure 1: Source:https://www.nature.com/articles/s41598-020-62922-y

The condition code embeddings are vectors of 288 dimensions. After the model is created, we use t-SNE to reduce the 288-dimensional vector to 2D. We then plot the diagnosis codes on a 2D map to gain visibility and better understanding of the embedding representation of the condition codes.

The disease prediction tasks are downstream tasks that use the same weights as the pre-training model created by the MLM. The input data is split into 80% for training and 20% for validation. For the next visit prediction, a random visit is selected for each patient, and all the prior visits are used as input. The conditions of the selected visit are converted to a multi-hot vector and used as a label that the task will attempt to predict. The random visit selection is performed from visit number 4 to N (where N is the number of visits for the patient)

For the next 6-month and 12-month predictions, we use a slightly different approach. First, we identify all visits that occur after visit number 4 and are at least 6 months (or 12 months for the next 12-month prediction) apart from the previous visit. If a patient has one or more visits that meet this condition, we randomly select one of them. Similar to the next visit prediction, we use all prior visits as input and the multi-hot vector representation of the selected visit's condition codes as a label.

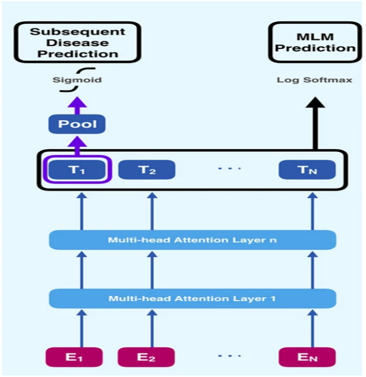

Subsequent Disease Prediction
MLM Prediction
Sigmoid
Log Softmax
Pool
T_1
T_2
...
T_N
Multi-head Attention Layer n
Multi-head Attention Layer 1
E_1
E_2
...
E_N

Figure 2: Source

3.2 Data set description

The core input data is the patient with demographics (age) at each doctor's visit. Each visit adds zero or more medical codes (condition or disease codes). Medical codes are represented by standard codes, which in our case is the ICD-10M coding standard. In EHR data, doctor visits don't follow a regular time pattern, so a timestamp is attached to each event to identify the order of events and length of time between events.

Instead of the original data source from CPRD, we used the outpatient synthetic data from medysin.ai[5]. The data consists of multiple CSV files following the Observational Medical Outcomes Partnership (OMOP) and Common Data Model (CDM). Patient data is obtained from the person file (person.csv), which contains a unique identifier (patient id) and demographics data (date of birth, gender, ethnicity, etc.). Hospital visits are stored in the visit_occurrence.csv file, which contains a unique visit id, visit date, person id, and other attributes such as location. Medical codes, such as diagnosis, procedures, drugs, etc., at each visit are stored in separate files, e.g. diagnoses codes are in the condition_occurrence.csv file.

In this project, we only used patient details (pid, dob), visit details (pid, visit date), and condition occurrence (visit date, diagnosis code). We created a pre-processing task to generate pickle files for BERT modeling. Key points are: (1) We calculated the frequency of occurrence of each condition across all patients and visits, and removed those infrequent conditions (3000 times of less). This brought the number of conditions from 6,997 to 296. We also added five special condition codes (CLS, SEP, PAD, MASK, and UNK), which made our vocab size 301, which is the same as BEHRT, but our prediction is limited to 296 conditions. (2) We removed patients with less than 5 visits, so we can have some medical history for next visit predictions.

3.2.1 Data statistics

Our data source consisted of 64,838 patients, 2.78 million unique patient visits, and 4.42 million condition codes in all visits. We removed patients with few number visits (less than 5). After pre-processing, the number of patients came down to 54,498, number of patient visits to 794,302, and the target condition codes to 296.

For predicting diagnoses after 6 (or 12) months from the last visit, a patient must have a visit that is dated 6 (or 12) months after the third visit. This further reduced the number of patients included in those downstream tasks to 26,187 for 6-months and 13,157 for 12-months prediction respectively.

3.2.2 Data representation

Electronic medical records are processed using language models, with diagnosis codes represented as words, visits as sentences, and a patient's medical history as a document. To facilitate this, two additional codes (”CLS” and ”SEP”) are added to the sequence to represent the start of a patient's medical code and to separate diagnosis codes between visits. In a language text, the order of words is critical. The order of condition codes for a patient's specific visit may not be as important. In this study, however, we sorted them in alpha-order.

A person's medical history is represented as a sequence of visits, with each visit having a unique sequence number used for positional encoding. The age sequence, unique to BEHRT, has the same length as condition tokens and stores the person's age at the time of the condition. A novel idea introduced by Transformers is the use of positional encoding to capture sequence information while avoiding sequential learning. Each condition code is assigned an order number, with conditions of a single visit having the same number. A segment encoding is added to differentiate conditions between two subsequent visits.

For example, for a patient with three visits [[D1, D2], [D1, D3], [D1]] Input tokens = ['CLS', 'D1', 'D2', 'SEP', 'D1', 'D3','D4', 'SEP', 'D1, 'SEP'] Position tokens = [0, 0, 0, 0, 1, 1, 1, 1, 2, 2] Segment token = [0, 0, 0, 0, 1, 1, 1, 1, 0, 0]

3.3 Model hyper-parameters

We used the same hyper-parameters as BEHRT to pre-train the data. Since our data size is smaller we tried training the model with different parameters, but the results were not better. We also tried to find the optimal parameters using Bayesian Optimization via the Optuna framework [6]. However, the Google Colab Pro environment was always crashing. The results of the manual parameter tuning is reported in section 4.3.

The following are the main hyper-parameters we used for the pre-training and mostly overlap with those used in the downstream tasks. The numbers in parenthesis are value we experimented, but either didn't produce or crashed the pre-training process.

- Number of attention heads = 12 (6): Lower number reduced the accuracy of MLM. .
- Number of hidden layers = 6 (3): Lower number reduced the accuracy of MLM.
- Hidden size = 288 (72). This correlates to the number of attention heads and hidden layers.
- Epochs = 50 for MLM and 20 for disease predictions) (10). Higher value produced better results.
- Batch size = 128 (256): Higher value caused memory crash.
- Learning rate = 3e-5 (1e-3): The lower learning rate produced better results.

3.4 Implementation

The system was implemented using python. The optimizer used for both MLM and disease prediction is BertAdam [3]. The BEHRT source code is publicly available at (https://github.com/deepmedicine/BEHRT).

Our source code is available in github. All the core functions and user executable actions are provided as Jupyter notebooks. Each notebook has basic information configuration information and installation of required dependencies. Additionally, the repository contains a landing page that provides general guidance. The link to shared Google Colab notebooks is also available in the repository's main page.

The pre-trained model used for next disease predictions is also located in github. This allows executing prediction tasks without running the pre-training tasks.

3.5 Computational requirements

We used Google Colab Pro with Python 3 Google Compute Engine back-end (GPU) which as 12.7GB system RAM and 15GB GPU RAM. We were able to run all the tasks with a batch size of 128. Larger batch sizes caused out of memory issues.

The pre-training task takes about 355 seconds per epoch. The total training for 50 epochs. The total process takes about 5 hours.

The next visit disease prediction task takes about 238 seconds per epoch or about 40 minutes for the default epochs of 10. The next 6 months and 12-months have similar computational need, but have a smaller data size.

4 Results

We conducted the pre-training task with 50 epochs, which produced the highest score in the MLM task. The model was saved and used for the downstream tasks. BEHRT used 100 epochs. The precision score of our model was 0.533 whereas BEHRT achieved 0.6597. For better comparability, we tried to run 100 epochs, but it didn't complete in Colab.

The APS and AUROC scores of the prediction tasks for BEHRT, Deepr, and RETAIN are presented comparatively below.

Model Name	**Next Visit**		**Next Visit-6months**	
	APS	**AUROC**	**APS**	**AUROC**
BEHRT	0.462	0.954	0.525	0.958
Deepr	0.360	0.942	0.393	0.943
RETAIN	0.382	0.921	0.413	0.928
Ours	0.508	0.874	0.461	0.830

4.1 Analysis

The accuracy of the MLM task was slight lower compared to BEHRT. We plotted the conditions codes in a 2D map. The purpose is to see whether similar conditions are shown closer in the map. We conducted visual inspections to see the overall distribution of condition codes and examining the name of the conditions that are close to each other. For example, Diabetes Mellitus Type I and Type II, anemia related diseases are very close. Also, Otitis media, Bronchitis, an upper respiratory diseases are shown closer.

We also computed the cosine similarity between all the vectors and selected the top 20 with a higher value. The result shows many similar conditions identified as similar. The top 10 are listed in section 4.2. The full list is available on the github repository.

BEHRT produced similar results. Additionally, the similarity computation was conducted to find the 10 most similar conditions for each of the 87 conditions that were selected due their prevalence (1%) in the population. BEHRT also conducted a validation by medical professionals.

Since our data sets are not the same, we didn't cross check for disease codes. Similar to BEHRT, we see some unrelated conditions showing closer, and that could be attributed to the extreme dimensionality reduction.

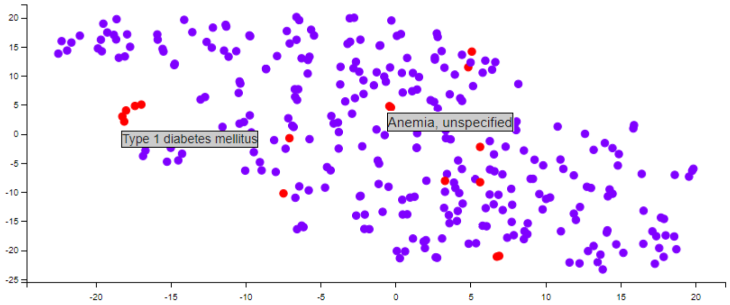


Figure 3: Disease mapping

4.2 Top 10 similar conditions identified by the model

1. Supervision of high risk pregnancy. Other maternal diseases classifiable elsewhere but complicating pregnancy, childbirth and the puerperium.

2. Supervision of high risk pregnancy Weeks of gestation.

3. Other hemorrhoids Polyp of colon

4. Neoplasm of uncertain behavior of skin Actinic keratosis

5. Subacute and chronic vulvitis Other specified noninflammatory disorders of vagina

6. Obstructive and reflux uropathy Calculus of kidney and ureter

7. Benign neoplasm of colon, rectum, anus and anal canal Diverticular disease of intestine

8. Other diseases of anus and rectum Hemorrhoids and perianal venous thrombosis

9. Superficial injury of head Other and unspecified injuries of head

10. Benign neoplasm of colon, rectum, anus and anal canal Hemorrhoids and perianal venous thrombosis.

4.3 Ablation study

We removed the age embedding and repeated the MLM tasks. The results were not that significantly different. The MLM precision scores are 0.460 (age in year is added), 0.455 (age in months), and 0.467 (age is not included). The comparative tests were conducted with 10 epochs each, and all the other parameters were keep the same.

5 Citations

References

[1] Edward Choi et al. "Medical Concept Representation Learning from Electronic Health Records and its Application on Heart Failure Prediction". In: (2017). arXiv: `1602.03686 [cs.LG]`.

[2] Edward Choi et al. *RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism.* 2017. arXiv: `1608.05745 [cs.LG]`.

[3] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`. URL: `https://aclanthology.org/N19-1423`.

[4] Emily Herrett et al. "Data Resource Profile: Clinical Practice Research Datalink (CPRD)". In: *International Journal of Epidemiology* 44.3 (June 2015), pp. 827–836. ISSN: 0300-5771. DOI: `10.1093/ije/dyv098`. eprint: `https://academic.oup.com/ije/article-pdf/44/3/827/14153119/dyv098.pdf`. URL: `https://doi.org/10.1093/ije/dyv098`.

[5] medysin.ai. *https://app.medisyn.ai/download.* `"https://app.medisyn.ai/download"`. [Online; Download synthetic EHR data].

[6] preferred networks. *https://optuna.org/.* `"https://optuna.org/"`. [Optuna Hyperparameter optimization framework].

[7] David MW Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061* (2020).

[8] Fatemeh Rahimian et al. "Predicting the risk of emergency admission with machine learning: Development and validation using linked electronic health records". In: *PLoS medicine* 15.11 (2018), e1002695.