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 [9] 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 [9].

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) [5] 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) [4] 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.

2 Scope of reproducibility

3 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 results closely match the BEHRT study and outperformed other compared approaches, i.e., Deepr [2] and RETAIN [2].

3.1 Data source

The core input data is the patient with its changing 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 are medical conditions that follow 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.

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 [6], which allowed us to download after registration. The data follows OMOP format.

3.2 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 (4 or less). The total number of unique condition codes is 6,977, but we removed those that are infrequent to align our prediction closer to BEHRT. As a result, conditions that showed less than 3,000 times were removed. 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.

3.3 Model 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 [7]. However, the Google Colab Pro environment was always crashing. The results of the manual parameter tuning is reported in section 5.3.

3.4 Source code

The source code of BEHRT is published on github [3], but there are some references to private codes for pre-processing the data. We made best effort to pre-process the data following the paper and the general input forms required by BERT.

One potential difference from the is the representation of age. The code implies using the age (in years), but the code that calculate age is not available. For many patients, the same age may be present multiple times due to repeated visits. To gain better temporal representation, we used the age in months as the full date of birth and visit dates are available in the data. The results, however, were not significantly different.

Another variation we introduce is the ordering of condition codes within a visit. As BERT is a language model ordering should be important. BEHRT does not explicitly mention this. We sorted the condition codes within a specific visit.

In summary, the limited quality volume of data and incomplete information limited our ability to reproduce the exact results of the research. Nevertheless, we achieved a comparable performance as shown in the results section.

4 Methodology

4.1 Data descriptions

Instead of the original data source from CPRD, we used the outpatient synthetic data from medysin.ai. The data consists of multiple CSV data 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 code 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 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 prediction. (3) We used 256 as the maximum sequence length because BERT uses fixed sequence length. The max length in our data set is 447. Therefore, we removed data for some patients and padded with a special character for those that have less data. We tried to train the model with 512 sequence length, but the tasks failed due to memory pressure.

4.2 Model descriptions

The system uses the BERT encoder architecture to generate a vector representation for each condition code based on its occurrence in patients 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), with about 4.5 million parameters in total.

We fine-tuned the system using four tasks: Masked Language Model (MLM) and Next Visit prediction, 6-months

prediction, and 12-months prediction. For MLM, we mask 15% of known condition codes and force the model to predict them. We evaluated the prediction accuracy. The resulting vector representation has 288 dimensions. We used t-SNE to plot diagnosis code in a map to gain more understanding of the embedding representation of condition codes.

4.3 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]

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

4.4 Implementation

The transformed input, i.e. the summation of all the embeddings, is feed to the BERT transformer-based architecture. First, the input is pre-trained using the MLM task where 86.5% of the condition codes were unchanged, 12% replaced with a mask, and the remaining 1.5% were changed with a 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 [8] of the pre-training MLM task is calculated over all patients and labels.

The disease prediction tasks are downstream tasks that uses 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 next visit prediction, for each patient, a random visit is selected. All the prior visits are used as input. The conditions of the selected visit are converted to multi-hot vector and used as a label that the task will attempt to predict. The random visit selection is performed from visit index 3 to N-1 (N = number visits of the patient).

For the next 6-months and 12-months predictions, we use a slightly different approach. First, we identify all visits that occur after index 4 and are at least 6 months (or 12 months for the next 12-months prediction) apart from the previous visit. If a patient has one or more visits that meet this condition, we randomly select a visit from the visits. 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 is used as a label.

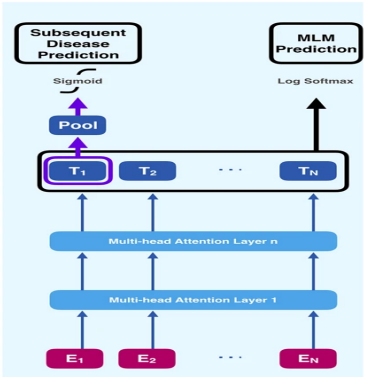

Figure 2: Source

The system was implemented using python. The optimizer used for both MLM and disease prediction is BertAdam [4].

The BEHRT source code is publicly available at (https://github.com/deepmedicine/BEHRT). We created

three Jupyter notebooks: pre-processing input csv files, BERT pre-training, and next visit prediction task. The synthetic data, notebooks, and supporting module files will be available on github as part of the final project submission.

4.5 Computational requirements

We used Google Colab with Python 3 Google Compute Engine back-end (GPU) which as 12.7GB system RAM and 15GB GPU RAM. We were able to both tasks with a batch size of 64. Larger batch sizes caused out of memory issues.

5 Results

The APS and AUROC scores of the model, BEHRT, Deepr, and RETAIN are presented 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

5.1 Analysis

The accuracy of the MLM task is almost the same as BEHRT. We also 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.

We also computed the cosine similarity between 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 5.2. The full list is available on the github repository.

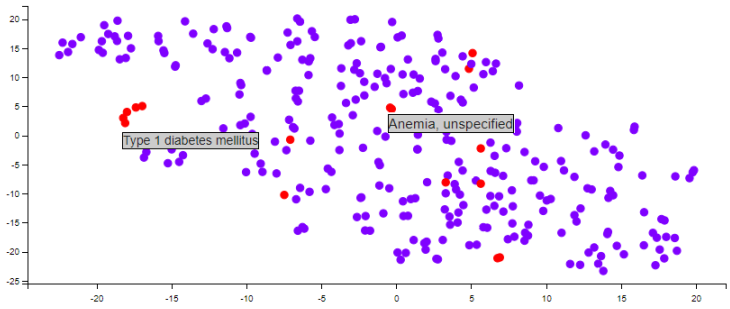

Figure 3: Disease mapping

5.2 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.

5.3 Ablation study and hyper-parameter tuning

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 is months), and 0.467 (age is not included). These tests were conducted with 10 epochs each, and all the parameters same as BEHRT.

The training was conducted with epoch 10, 30, and 50. Epoch 50 achieved the highest score for the MLM task (0.533). This model was saved and used for the downstream tasks.

We also experimented with different model parameters, such as number of attention heads, hidden size, and sequence length. The best performance with no system crash was obtained when using the same hyper-parameters as BEHRT.

6 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] deepmedicine. *https://github.com/deepmedicine/BEHRT*. "https://github.com/deepmedicine/BEHRT". [BEHRT code].

[4] 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.

[5] 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.

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

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

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

[9] 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.