CS598 Project: Deep Learning for Healthcare

Deepa N Veeravalli & Sudhir Koundinya Nagesh
{deepan2, sudhirk2}@illinois.edu

Group ID: 61
Paper ID: 17

Introduction

We have chosen "Pre-training of Graph Augmented Transformers for Medical Recommendation" paper, authored by Junyuan Shang, Tengfei Ma, Cao Xiao and Jimeng Sun, for our final project. The authors propose G-BERT graph learning model for medication recommendation.

The advances in Deep Learning technologies, availability of resources and data have provided great opportunities in the field of medical recommendation. The authors argue that existing medication recommendation models often learn the representations from the longitudinal EHR data from a small number of patients with multiple visits, ignoring patients with single visit (*selection bias*) and ignore the complex hierarchical relationships among diagnosis, medications, symptoms, and diseases (lack of hierarchical knowledge), which can lead to suboptimal recommendations. To address these issues and enhance the prediction and interpretability, they propose G-BERT graph learning model. The proposed model leverages medical knowledge encoded in large-scale medical corpora and single visit patient information to pre-train the GAT model to learn the underlying structure of medical concepts and relationships. The purpose of pre-training is mainly to leverage patient record with single visit and provide model trainings with good initializations.

The pre-trained GAT model is then fine-tuned on patient medical records with multiple visits, to personalize the medication recommendation for each patient. The model can capture not only the direct relationships between medications and symptoms but also the indirect relationships between medications through shared diseases or symptoms.

Scope of Reproducibility

The authors have implemented G-BERT model for Medical Recommendation with GAT (a popular GNN architecture) using MIMIC-III dataset. As part of the reproduction study, we would like to implement the same model with GATv2 architecture, to reproduce the original results and achieve Jaccard Similarity score >= 0.35, PR-AUC >= 0.50 and F1 score >= 0.50 using the same MIMIC-III dataset the authors have used for their study. The authors of GATv2 claim that this architecture, a dynamic graph variant which is more expressive than GAT. We plan to perform few ablation studies using the same dataset as well.

Method

The overall framework of G-BERT consists of three main parts: Ontology Embedding, BERT and fine-tuned classifier, which is described below.

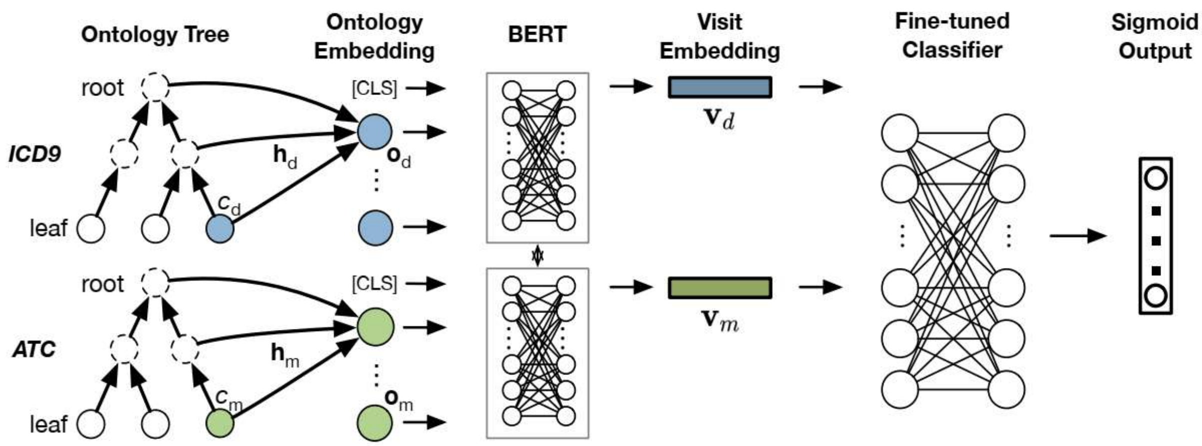

Fig 1: G-BERT Framework

The specific enhancements discussed are:

- Pre-training to leverage more data - adapt the framework of BERT and pre-train the model on each visit of the EHR data to leverage the single visit data that were not fit for training in earlier medication recommendation models.
- Medical ontology embedding with graph neural networks
- Ontology embedding for medical code laid in leaf nodes by cooperating ancestors' information based on graph attention networks.
- Next, input set of diagnosis and medication ontology embedding separately to shared weight BERT which is pre-trained using single visit data.
- Finally, concatenate the mean of all previous visit embeddings and the last visit embedding as input and fine-tune the prediction layers for medication recommendation tasks.

The above-mentioned approach is interesting and innovative because it addresses the key limitations of existing medication recommendation models, such as selection bias, the lack of consideration of complex hierarchical relationships among diagnosis, medications, symptoms, and diseases and increases interpretability. It provides more personalized and effective medication recommendations for patients, leading to improved patient outcomes.

As part of this reproduction study, we implemented G-BERT model with GATv2 architecture. GATv2 is a dynamic graph variant, and it is supposed to be more expressive than GAT. The authors of GATv2 claim that GAT uses a static attention mechanism and there are simple graph problems that GAT cannot express, and this static attention hinders GAT from even fitting the training data.

We are using the MIMIC-III synthetic data (preprocessed pickle files) made available in the git repository (https://github.com/jshang123/G-Bert). The statistics of the data is provided below.

Stats	Single-Visit	Multi-Visit
# of patients	30,745	6,350
avg # of visits	1.00	2.36
avg # of dx	39	10.51
avg # of rx	52	8.80
# of unique dx	1,997	1,958
# of unique rx	323	145

Table 1: Data Statistics (dx for diagnosis and rx for medication)

Our implementation is a mix of new code and reuse of the existing code. The author's documentation was missing specific version for some of the libraries used. Hence, we had to determine the right version of the libraries before we could run the base line code (provided in the git) end to end on our local machines. Our local machines have Ubuntu 22.04 with no GPU. To pre-train and run the model for 1 epoch, it takes about 45 mins on our local machines. Because of this, we are not able to run for a greater number of epochs. We definitely need GPU with at least 12 GB of memory. We will use Google cloud environment for more extensive training, testing (including hyper-parameter testing) and ablation studies. As part of ablation, we plan to use the medical code embedding without ontology as input, with/without pre-training and compare the performance.

Result

Below table provides the performance comparison of G-BERT with GAT and GATv2 architecture for 1 epoch. With just 1 epoch we cannot draw any conclusions on the result. Because of the hardware limitations on our local machines, we will use Google cloud environment for extensive training, testing (including hyper-parameter testing) and ablation studies.

	Jaccard	PR-AUC	F1
G-BERT (GAT)	0.3857	0.6032	0.5465
G-BERT (GATv2)	0.3778	0.5988	0.5392

Table 2: Performance Comparison

Conclusion

In summary, we were able to run the G-BERT base line (GAT) and newly implemented code (GATv2) on our local machines for 1 epoch using MIMIC-III synthetic dataset and compare the results. As there are hardware limitations on our local machines, we will use Google cloud environment for more extensive training and testing (including hyper-parameter testing) and then compare the results to draw final conclusion.

References

Textbook: Introduction to Deep Learning for Healthcare by Cao Xiao and Jimeng Sun

Original Paper: https://www.ijcai.org/proceedings/2019/0825.pdf

GATv2 Paper: https://arxiv.org/pdf/2105.14491.pdf

Asif Qamar*, Alumni UIUC, SupportVectors AI Labs.*