Pre-training of Graph Augmented Transformers for Medication Recommendation

Junyuan Shang[1,3] , **Tengfei Ma**[2] , **Cao Xiao**[1] and **Jimeng Sun**[3]

[1]Analytics Center of Excellence, IQVIA, Cambridge, MA, USA
[2]IBM Research AI, Yorktown Heights, NY, USA
[3]Georgia Institute of Technology, Atlanta, GA, USA

junyuan.shang@iqvia.com, Tengfei.Ma1@ibm.com, cao.xiao@iqvia.com, jsun@cc.gatech.edu

Abstract

Medication recommendation is an important healthcare application. It is commonly formulated as a temporal prediction task. Hence, most existing works only utilize longitudinal electronic health records (EHRs) from a small number of patients with multiple visits ignoring a large number of patients with a single visit (*selection bias*). Moreover, important *hierarchical knowledge* such as diagnosis hierarchy is not leveraged in the representation learning process. To address these challenges, we propose G-BERT, a new model to combine the power of Graph Neural Networks (GNNs) and BERT (Bidirectional Encoder Representations from Transformers) for medical code representation and medication recommendation. We use GNNs to represent the internal hierarchical structures of medical codes. Then we integrate the GNN representation into a transformer-based visit encoder and pre-train it on EHR data from patients only with a single visit. The pre-trained visit encoder and representation are then fine-tuned for downstream predictive tasks on longitudinal EHRs from patients with multiple visits. G-BERT is the first to bring the language model pre-training schema into the healthcare domain and it achieved state-of-the-art performance on the medication recommendation task.

1 Introduction

The availability of massive electronic health records (EHR) data and the advances of deep learning technologies have provided unprecedented resource and opportunity for predictive healthcare, including the computational medication recommendation task. A number of deep learning models were proposed to assist doctors in making medication recommendation [Xiao *et al.*, 2018a; Shang *et al.*, 2019; Baytas *et al.*, 2017; Choi *et al.*, 2018; Ma *et al.*, 2018]. They often learn representations for medical entities (e.g., patients, diagnosis, medications) from patient EHR data, and then use the learned representations to predict medications that are suited to the patient's health condition.
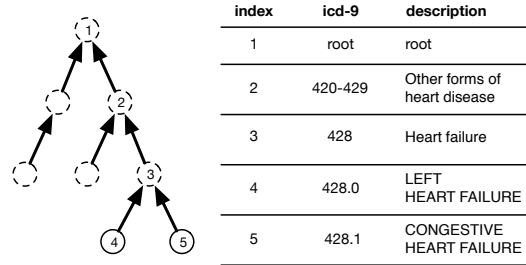

Figure 1: Graphical illustration of ICD-9 Ontology.

To provide effective medication recommendation, it is important to learn accurate representation of medical codes. Despite that various considerations were handled in previous works for improving medical code representations [Ma *et al.*, 2018; Baytas *et al.*, 2017; Choi *et al.*, 2018], there are two limitations with the existing work:

1. **Selection bias:** Data that do not meet training data criteria are discarded before model training. For example, a large number of patients who only have one hospital visit were discarded from training in [Shang *et al.*, 2019].

2. **Lack of hierarchical knowledge:** For medical knowledge such as diagnosis code ontology (Figure. 1), their internal hierarchical structures were rarely embedded in their original graph form when incorporated into representation learning.

To mitigate the aforementioned limitations, we propose G-BERT that combines the pre-training techniques and graph neural networks for better medical code representation and medication recommendation. G-BERT is enabled and demonstrated by the following technical contributions:

1. *Pre-training to leverage more data:* Pre-training techniques, such as ELMo [Peters *et al.*, 2018], OpenAI GPT [Radford *et al.*, 2018] and BERT [Devlin *et al.*, 2018], have demonstrated a notably good performance in various natural language processing tasks. These techniques generally train language models from unlabeled data, and then adapt the derived representations to different tasks by either feature-based (e.g. ELMo) or fine-tuning (e.g. OpenAI GPT, BERT) methods. We developed a new pre-training method based on BERT for

pre-training on each visit of EHR so that the data with only one hospital visit can also be utilized. We revised BERT to fit EHR data in both input and pre-training objectives. To our best knowledge, `G-BERT` is the first model that leverages Transformers and language model pre-training techniques in healthcare domain. Compared with other supervised models, `G-BERT` can utilize discarded/unlabeled data more efficiently.

2. *Medical ontology embedding with graph neural networks:* We enhance the representation of medical codes via learning medical ontology embedding for each medical codes with graph neural networks. We then input the ontology embedding into a multi-layer Transformer [Vaswani *et al.*, 2017] for BERT-style pre-training and fine-tuning.

2 Related Work

Medication Recommendation. Medication Recommendation can be categorized into instance-based and longitudinal recommendation methods [Shang *et al.*, 2019]. Instance-based methods focus on current health conditions. Among them, Leap [Zhang *et al.*, 2017] formulates a multi-instance multi-label learning framework and proposes a variant of sequence-to-sequence model based on content-attention mechanism to predict combination of medicines given patient's diagnoses. Longitudinal-based methods leverage the temporal dependencies among clinical events, see [Choi *et al.*, 2016; Xiao *et al.*, 2018b; Lipton *et al.*, 2015]. Among them, RETAIN [Choi *et al.*, 2016] uses a two-level neural attention model to detect influential past visits and significant clinical variables within those visits for improved medication recommendation.

Pre-training Techniques. The goal of pre-training techniques is to provide model training with good initializations. Pre-training has been shown extremely effective in various areas such as image classification [Hinton *et al.*, 2006] and machine translation [Ramachandran *et al.*, 2016]. The unsupervised pre-training can be considered as a regularizer that supports better generalization from the training dataset [Erhan *et al.*, 2010]. Recently, language model pre-training techniques such as [Peters *et al.*, 2018; Radford *et al.*, 2018; Devlin *et al.*, 2018] have shown to largely improve the performance on multiple NLP tasks. As the most widely used one, BERT [Devlin *et al.*, 2018] builds on the Transformer [Vaswani *et al.*, 2017] architecture and improves the pre-training using a masked language model for bidirectional representation. In this paper, we adapt the framework of BERT and pre-train our model on each visit of the EHR data to leverage the single-visit data that were not fit for training in other medication recommendation models.

Graph Neural Networks (GNN). GNNs are neural networks that learn node or graph representations from graph-structured data. Various graph neural networks have been proposed to encode the graph-structure information, including graph convolutional neural networks (GCN) [Kipf and

Welling, 2017], message passing networks (MPNN) [Gilmer *et al.*, 2017], graph attention networks (GAT) [Velickovic *et al.*, 2017]. GNNs have already been demonstrated useful on EHR modeling [Choi *et al.*, 2017; Shang *et al.*, 2019]. GRAM [Choi *et al.*, 2017] represented a medical concept as a combination of its ancestors in the medical ontology using an attention mechanism. It's different from `G-BERT` from two aspects as described in Section 4.2. Another work worth mentioning is GAMENet [Shang *et al.*, 2019], which also used graph neural network to assist the medication recommendation task. However, GAMENet has a different motivation which results in using graph neural networks on drug-drug-interaction graphs instead of medical ontology.

3 Problem Formalization

Definition 1 (Longitudinal Patient Records). In longitudinal EHR data, each patient can be represented as a sequence of multivariate observations: $\mathcal{X}^{(n)} = \{\mathcal{X}_1^{(n)}, \mathcal{X}_2^{(n)}, \cdots, \mathcal{X}_{T^{(n)}}^{(n)}\}$ where $n \in \{1, 2, \ldots, N\}$, N is the total number of patients; $T^{(n)}$ is the number of visits of the nth patient. Here we choose two main medical code to represent each visit $\mathcal{X}_t = \mathcal{C}_d^t \cup \mathcal{C}_m^t$ of a patient which is a union set of corresponding diagnoses codes $\mathcal{C}_d^t \subset \mathcal{C}_d$ and medications codes $\mathcal{C}_m^t \subset \mathcal{C}_m$. For simplicity, we use $\mathcal{C}_*^t$ to indicate the unified definition for different type of medical codes and drop the superscript (n) for a single patient whenever it is unambiguous. $\mathcal{C}_*$ denotes the medical code set and $|\mathcal{C}_*|$ the size of the code set. $c_* \in \mathcal{C}_*$ is the medical code.

Definition 2 (Medical Ontology). Medical codes are usually categorized according to a tree-structured classification system such as ICD-9 ontoloy for diagnosis and ATC ontology for medication. We use $\mathcal{O}_d, \mathcal{O}_m$ to denote the ontology for diagnosis and medication. Similarly, we use $\mathcal{O}_*$ to indicate the unified definition for different type of medical codes. In detial, $\mathcal{O}_* = \overline{\mathcal{C}_*} \cup \mathcal{C}_*$ where $\overline{\mathcal{C}_*}$ denotes the codes excluding leaf codes. For simplicity, we define two function $pa(\cdot), ch(\cdot)$ which accept target medical code and return ancestors' code set and direct child code set.

Problem Definition (Medication Recommendation). Given diagnosis codes $\mathcal{C}_d^t$ of the visit at time t, patient history $\mathcal{X}_{1:t} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_{t-1}\}$, we want to recommend multiple medications by generating multi-label output $\hat{\boldsymbol{y}}_t \in \{0, 1\}^{|\mathcal{C}_m|}$.

4 Method

The overall framework of `G-BERT` is described in Figure 2. `G-BERT` first derives the initial embedding of medical codes from medical ontology using graph neural networks. Then, in order to fully utilize the rich EHR data, `G-BERT` constructs an adaptive BERT model on the discarded single-visit data for visit representation. Finally we add a prediction layer and fine-tune the model in the medication recommendation task. In the following we will describe `G-BERT` in detail. But firstly, we give a brief background of BERT especially for the two pre-training objectives which will be later adapted to EHR data in Section 4.3.
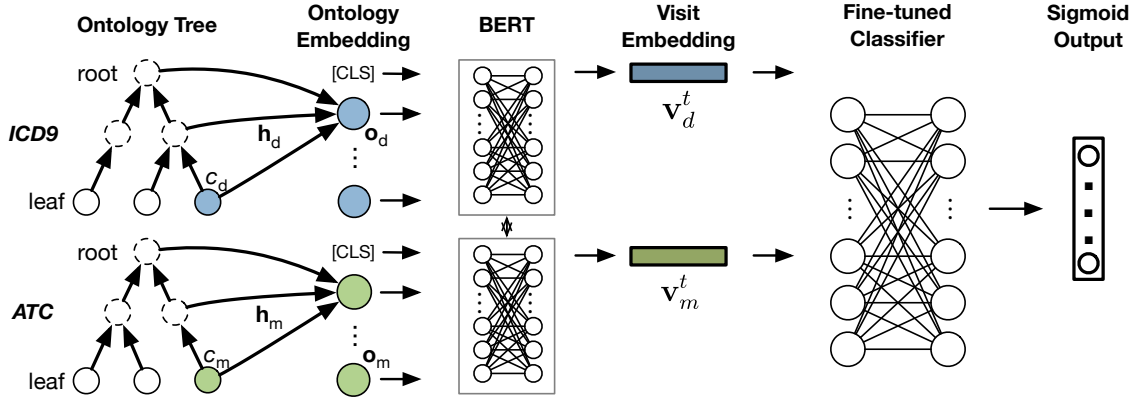
Figure 2: The framework of G-BERT. It consists of three main parts: ontology embedding, BERT and fine-tuned classifier. Firstly, we derive ontology embedding for medical code laid in leaf nodes by cooperating ancestors information by Eq. 1 and 2 based on graph attention networks (Eq. 3, 4). Then we input set of diagnosis and medication ontology embedding separately to shared weight BERT which is pretrained using Eq. 6, 7, 8. Finally, we concatenate the mean of all previous visit embeddings and the last visit embedding as input and fine-tune the prediction layers using Eq. 10 for medication recommendation tasks.

Notation	Description
$\mathcal{X}^{(n)}$	longitudinal observations for n-th patient
$\mathcal{C}_d, \mathcal{C}_m$	diagnoses and medications codes set
$\mathcal{O}_d, \mathcal{O}_m$	diagnoses and medications codes ontology
$\overline{\mathcal{C}_*} \subset \mathcal{O}_*$	non-leaf medical codes of type $*$
$c_* \in \mathcal{O}_*$	single medical code of type $*$
$pa(c_*)$	function retrieve c_*' ancestors' code
$ch(c_*)$	function retrieve c_*' direct children's code
$\boldsymbol{W}_e \in \mathbb{R}^{|\mathcal{O}_*| \times d}$	initial medical embedding matrix in $\mathcal{O}_*$
$\boldsymbol{H}_e \in \mathbb{R}^{|\mathcal{O}_*| \times d}$	enhanced medical embeddings in **stage 1**
$\boldsymbol{o}_{c_*} \in \mathbb{R}^d$	ontology embedding in **stage 2**
$\alpha_{i,j}^k$	k-th attention between nodes i, j
$\boldsymbol{W}^k \in \mathbb{R}^{d \times d}$	k-th weight matrix applied to each node
$g(\cdot, \cdot, \cdot)$	graph aggregator function
$\boldsymbol{v}_*^t$	t-th visit embedding of type $*$
$\hat{\boldsymbol{y}}_t \in \{0,1\}^{|\mathcal{C}_*|}$	multi-label prediction

Table 1: Notations used in G-BERT.

4.1 Background of BERT

Based on a multi-layer Transformer encoder [Vaswani *et al.*, 2017] (The transformer architecture has been ubiquitously used in many sequence modeling tasks recently, so we will not introduce the details here), BERT is pre-trained using two unsupervised tasks:

- Masked Language Model. Instead of predicting words based on previous words, BERT randomly selects words to mask out and then predicts the original vocabulary IDs of the masked words from their (bidirectional) context.

- Next Sentence Prediction. Many of BERT's downstream tasks are predicting the relationships of two sentences, thus in the pre-training phase, BERT has am a binary sentence prediction task to predict whether one sentence is the next sentence of the other.

A typical input to BERT is as follows ([Devlin *et al.*, 2018]):

Input = [CLS] the man went to [MASK] store

[SEP] he bought a gallon [MASK] milk [SEP]
Label = IsNext

where [CLS] is the first token of each sentence pair to represent the special classification embedding, i.e. the final state of this token is used as the aggregated sequence representation for classification tasks; [SEP] is used to separate two sentences; [MASK] is used to mask out the predicted words in the masked language model. Using this form, these inputs facilitate the two tasks described above, and they will also be used in our method description in the following section.

4.2 Input Representation

The G-BERT model takes medical codes' ontology embeddings as input, and obtains intermediate representations from a Transformer encoder as the visit embeddings. It is then pre-trained on EHR from patients who only have one hospital visit. The derived encoder and visit embedding will be fed into a classifier and fine-tuned to make predictions.

Ontology Embedding
We constructed ontology embedding from diagnosis ontology $\mathcal{O}_d$ and medication ontology $\mathcal{O}_m$. Since the medical codes in raw EHR data can be considered as leaf nodes in these ontology trees, we can enhance the medical code embedding using graph neural networks (GNNs) to integrate the ancestors' information of these codes. Here we perform a two-stage procedure with a specially designed GNN for ontology embedding.

To start, we assign an initial embedding vector to every medical code $c_* \in \mathcal{O}_*$ with a learnable embedding matrix $\boldsymbol{W}_e \in \mathbb{R}^{|\mathcal{O}_*| \times d}$ where d is the embedding dimension.

Stage 1. For each non-leaf node $c_* \in \overline{\mathcal{C}_*}$, we obtain its enhanced medical embedding $\boldsymbol{h}_{c_*} \in \mathbb{R}^d$ as follows:

$$\boldsymbol{h}_{c_*} = g(c_*, ch(c_*), \boldsymbol{W}_e) \tag{1}$$

where $g(\cdot, \cdot, \cdot)$ is an aggregation function which accepts the target medical code c_*, its direct child codes $ch(c_*)$ and initial embedding matrix. Intuitively, the aggregation function can pass and fuse information in target node from its direct children which result in the more related embedding of ancestor' code to child codes' embedding.

Stage 2. After obtaining enhanced embeddings, we pass the enhance embedding matrix $\boldsymbol{H}_e \in \mathbb{R}^{|\mathcal{O}_*| \times d}$ back to get ontology embedding for leaf codes $c_* \in \mathcal{C}_*$ as follows:

$$\boldsymbol{o}_{c_*} = g(c_*, pa(c_*), \boldsymbol{H}_e) \qquad (2)$$

where $g(\cdot, \cdot, \cdot)$ accepts ancestor codes of target medical code c_*. Here, we use $pa(c_*)$ instead of $ch(c_*)$, since utilizing the ancestors' embedding can indirectly associate all medical codes instead of taking each leaf code as independent input.

The option for the aggregation function $g(\cdot, \cdot, \cdot)$ is flexible, including *sum*, *mean*. Here we choose the one from graph attention networks (GAT) [Velickovic *et al.*, 2017], which has shown efficient embedding learning ability on graph-structured tasks, e.g., node classification and link prediction. In particular, we implement the aggregation function $g(\cdot, \cdot, \cdot)$ as follows (taking stage 2 for an example):

$$g(c_*, p(c_*), \boldsymbol{H}_e) = \overset{K}{\underset{k=1}{\big\|}} \sigma \left(\sum_{j \in \{c_*\} \cup pa(c_*)} \alpha_{i,j}^k \boldsymbol{W}^k \boldsymbol{h}_j \right) \quad (3)$$

where $\|$ represents concatenation which enables the multi-head attention mechanism, σ is a nonlinear activation function, $\boldsymbol{W}^k \in \mathbb{R}^{m \times d}$ is the weight matrix for input transformation, and $\alpha_{i,j}^k$ are the corresponding k-th normalized attention coefficients computed as follows:

$$\alpha_{i,j}^k = \frac{\exp\left(\text{LeakyReLU}(\boldsymbol{a}^\intercal [\boldsymbol{W}^k \boldsymbol{h}_i || \boldsymbol{W}^k \boldsymbol{h}_j])\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}(\boldsymbol{a}^\intercal [\boldsymbol{W}^k \boldsymbol{h}_i || \boldsymbol{W}^k \boldsymbol{h}_k])\right)}$$
$$(4)$$

where $\boldsymbol{a} \in \mathbb{R}^{2m}$ is a learnable weight vector and LeakyReLU is a nonlinear function. (we assume $m = d/K$).

As shown in Figure 2, we construct ICD-9 tree for diagnosis and ATC tree for medication using the same structure. Here the direction of arrow shows the information flow where ancestor nodes can get information from their direct children (in **stage 1**) and similarly leaf nodes can get information from their connected ancestors (in **stage 2**).

It is worth mentioning that our graph embedding method on medical ontology is different from GRAM [Choi *et al.*, 2017] from the following two aspects:

1. *Initialization*: we initialize all the node embeddings from a learnable embedding matrix, while GRAM learns them using Glove from the co-occurrence information.

2. *Updating*: we develop a two-step updating function for both leaf nodes and ancestor nodes; while in GRAM, only the leaf nodes are updated (as a combination of their ancestor nodes and themselves).

Visit Embedding

Similar to BERT, we use a multi-layer Transformer architecture [Vaswani *et al.*, 2017] as our visit encoder. The model takes the ontology embedding as input and derive visit embedding $\boldsymbol{v}_*^t \in \mathbb{R}^d$ for a patient at t-th visit:

$$\boldsymbol{v}_*^t = \text{Transformer}(\{[CLS]\} \cup \{\boldsymbol{o}_{c_*}^t | c_* \in \mathcal{C}_*^t\})[0] \qquad (5)$$

where *[CLS]* is a special token as in BERT. It is put in the first position of each visit of type $*$ and its final state can be used as the representation of the visit. Intuitively, it is more reasonable to use Transformers as encoders (multi-head attention based architecture) than RNN or mean/sum to aggregate multiple medical embedding for visit embedding since the set of medical codes within one visit is not ordered.

It is worth noting that our Transformer encoder is different from the original one in the position embedding part. Position embedding, as an important component in Transformers and BERT, is used to encode the position and order information of each token in a sequence. However, one big difference between language sentences and EHR sequences is that the medical codes within the same visit do not generally have an order, so we remove the position embedding in our model.

4.3 Pre-training

We adapted the original BERT model to be more suitable for our data and task. In particular, we pre-train the model on each EHR visit (within both single-visit EHR sequences and multi-visit EHR sequences). We modified the input and pre-training objectives of the BERT model: (1) For the input, we built the Transformer encoder on the GNN outputs, i.e. ontology embeddings, for visit embedding. For the original EHR sequence, it means essentially we combine the GNN model with a Transformer to become a new integrated encoder. In addition, we removed the position embedding as we explained before. (2) As for the pre-training procedures, we modified the original pre-training tasks i.e., Masked LM (language model) task and Next Sentence prediction task to **self-prediction task** and **dual-prediction task**. The idea to conduct these tasks is to make the visit embedding absorb enough information about what it is made of and what it is able to predict.

Thus, for the **self-prediction task**, we want the visit embedding v_*^1 to recover what it is made of, i.e., the input medical codes $\mathcal{C}_*^t$ for each visit as follows:

$$\mathcal{L}_{se}(\boldsymbol{v}_*^1, \mathcal{C}_*^1) = -\log p(\mathcal{C}_*^1 | \boldsymbol{v}_*^1)$$
$$= -\sum_{c \in \mathcal{C}_*^1} \log p(c_*^1 | \boldsymbol{v}_*^1) + \sum_{c \in \mathcal{C}_* \setminus \mathcal{C}_*^1} \log p(c_*^1 | \boldsymbol{v}_*^1) \quad (6)$$

we minimize the binary cross entropy loss $\mathcal{L}_s$, and in practise, Sigmoid($f(\boldsymbol{v}_*)$) should be transformed by applying a fully connected neural network $f(\cdot)$ with one hidden layer. With an analogy to the Masked LM task in BERT, we also used specific symbol [MASK] to randomly replace the original medical code $c_* \in \mathcal{C}_*^1$. So there are 15% codes in $\mathcal{C}_*$ which will be replaced randomly and the model should have the

ability to predict the masked code based on others.

Likewise, for the **dual-prediction task**, since the visit embedding $\boldsymbol{v}_*$ carries the information of medical codes of type $*$, we can further expect it has the ability to do more task-specific prediction as follows:

$$\mathcal{L}_{du} = -\log p(\mathcal{C}_d^1|\boldsymbol{v}_m^1) - \log p(\mathcal{C}_m^1|\boldsymbol{v}_d^1) \qquad (7)$$

where we use the same transformation function $\mathrm{Sigmoid}(f_1(\boldsymbol{v}_m^1))$, $\mathrm{Sigmoid}(f_2(\boldsymbol{v}_d^1))$ with different weight matrix to transform the visit embedding and optimize the binary cross entropy loss $\mathcal{L}_{du}$ expanded same as $\mathcal{L}_{se}$ in Eq. 6. This is a direct adaptation of the next sentence prediction task. In BERT, the next sentence prediction task facilitates the prediction of sentence relations, which is a common task in NLP. However, in healthcare, most predictive tasks do not have a sequence pair to classify. Instead, we are often interested in predicting unknown disease or medication codes of the sequence. For example, in medication recommendation, we want to predict multiple medications given only the diagnosis codes. Inversely, we can also predict unknown diagnosis given the medication codes.

Thus, our final pre-training optimization objective can simply be the combination of the aforementioned losses, as shown in Eq. 8. It is used to train on EHR data from all patients who only have one hospital visits..

$$\mathcal{L}_{pr} = \frac{1}{N} \sum_{n=1}^{N} ((\mathcal{L}_{se}(\boldsymbol{v}_d^{1,(n)}, \mathcal{C}_d^{1,(n)}) + \mathcal{L}_{se}(\boldsymbol{v}_m^{1,(n)}, \mathcal{C}_m^{1,(n)})$$
$$+ \mathcal{L}_{du}^{(n)})) \qquad (8)$$

4.4 Fine-tuning

After obtaining pre-trained visit representation for each visit, for a prediction task on a multi-visit sequence data, we aggregate all the visit embedding and add a prediction layer for the medication recommendation task. To be specific, from pre-training on all visits, we have a pre-trained Transformer encoder, which can then be used to get the visit embedding $\boldsymbol{v}_*^\tau$ at time τ. The known diagnosis codes $\mathcal{C}_d^t$ at the prediction time t is also represented using the same model as $\boldsymbol{v}_*^t$. Concatenating the mean of previous diagnoses visit embeddings and medication visit embeddings, also the last diagnoses visit embedding, we built an MLP based prediction layer to predict the recommended medication codes as in Equation 9.

$$\boldsymbol{y}_t = \mathrm{Sigmoid}(\boldsymbol{W}_1[(\frac{1}{t}\sum_{\tau<t}\boldsymbol{v}_d^\tau)||(\frac{1}{t}\sum_{\tau<t}\boldsymbol{v}_m^\tau)||\boldsymbol{v}_d^t] + b) \quad (9)$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{|\mathcal{C}_m|\times 3d}$ is a learnable transformation matrix.

Given the true labels $\hat{\boldsymbol{y}}_t$ at each time stamp t, the loss function for the whole EHR sequence (i.e. a patient) is

$$\mathcal{L} = -\frac{1}{T-1}\sum_{t=2}^{T}(\boldsymbol{y}_t^\intercal\log(\hat{\boldsymbol{y}}_t) + (1 - \boldsymbol{y}_t^\intercal)\log(1 - \hat{\boldsymbol{y}}_t)) \quad (10)$$

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 2: Statistics of the Data (dx for diagnosis, rx for medication).

5 Experiment

5.1 Experimental Setting

Data

We used EHR data from MIMIC-III [Johnson *et al.*, 2016] and conducted all our experiments on a cohort where patients have more than one visit. We utilize data from patients with both single visit and multiple visits in the training dataset as pre-training data source (multi-visit data are split into visit slices). In this work, we transform the drug coding from NDC to ATC Third Level for using the ontology information. The statistics of the datasets are summarized in Table 2.

Baselines

We compared G-BERT[1] with the following baselines. All methods are implemented in PyTorch [Paszke *et al.*, 2017] and trained on an Ubuntu 16.04 with 8GB memory and Nvidia 1080 GPU.

1. **Logistic Regression (LR)** is logistic regression with L1/L2 regularization. Here we represent sequential multiple medical codes by sum of multi-hot vector of each visit. Binary relevance technique [Luaces *et al.*, 2012] is used to handle multi-label output.

2. **LEAP** [Zhang *et al.*, 2017] is an instance-based medication combination recommendation method which formalizes the task in multi-instance and multi-label learning framework. It utilizes a encoder-decoder based model with attention mechanism to build complex dependency among diseases and medications.

3. **RETAIN** [Choi *et al.*, 2016] makes sequential prediction of medication combination and diseases prediction based on a two-level neural attention model that detects influential past visits and clinical variables within those visits.

4. **GRAM** [Choi *et al.*, 2017] injects domain knowledge (ICD9 Dx code tree) to **tanh** via attention mechanism.

5. **GAMENet** [Shang *et al.*, 2019] is the method to recommend accuracy and safe medication based on memory neural networks and graph convolutional networks by leveraging EHR data and Drug-Drug Interaction (DDI) data source. For fair comparison, we use a variant of GAMENet without DDI knowledge and procedure codes as input renamed as GAMENet⁻.

6. **G-BERT** is our proposed model which integrated the GNN representation into Transformer-based visit encoder with pre-training on single-visit EHR data.

---

[1] https://github.com/jshang123/G-Bert

We also evaluated 3 G-BERT variants for model ablation.

1. **G-BERT**$_{G-,P-}$: We directly use medical embedding without ontology information as input and initialize the model's parameters without pre-training.

2. **G-BERT**$_{G-}$: We directly use medical embedding without ontology information as input with pre-training.

3. **G-BERT**$_{P-}$: We use ontology information to get ontology embedding as input and initialize the model's parameters without pre-training.

Metrics

To measure the prediction accuracy, we used Jaccard Similarity Score (Jaccard), Average F1 (F1) and Precision Recall AUC (PR-AUC). Jaccard is defined as the size of the intersection divided by the size of the union of ground truth set $Y_t^{(k)}$ and predicted set $\hat{Y}_t^{(k)}$.

$$\text{Jaccard} = \frac{1}{\sum_k^N \sum_t^{T_k} 1} \sum_k^N \sum_t^{T_k} \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|Y_t^{(k)} \cup \hat{Y}_t^{(k)}|}$$

where N is the number of patients in test set and T_k is the number of visits of the kth patient.

Implementation Details

We randomly divide the dataset into training, validation and testing set in a $0.6 : 0.2 : 0.2$ ratio. For G-BERT, the hyperparameters are adjusted on evaluation set: (1) GAT part: input embedding dimension as 75, number of attention heads as 4; (2) BERT part: hidden dimension as 300, dimension of position-wise feed-forward networks as 300, 2 hidden layers with 4 attention heads for each layer. Specially, we alternated the pre-training with 5 epochs and fine-tuning procedure with 5 epochs for 15 times to stabilize the training procedure.

For LR, we use the grid search over typical range of hyperparameter to search the best hyperparameter values which result in L1 norm penalty with weight as 1.1. For deep learning models, we implemented RNN using a gated recurrent unit (GRU) [Cho et al., 2014] and utilize dropout with a probability of 0.4 on the output of embedding. We test several embedding choice for baseline methods and determine the dimension for medical embedding as 300 and thershold for final prediction as 0.3 for better performance. Training is done through Adam [Kingma and Ba, 2014] at learning rate 5e-4. We fix the best model on evaluation set within 100 epochs and report the performance in test set.

5.2 Results

Table. 3 compares the performance on the medication recommendation task. For variants of G-BERT, G-BERT$_{G-,P-}$ performs worse compared with G-BERT$_{G-}$ and G-BERT$_{P-}$ which demonstrate the effectiveness of using ontology information to get enhanced medical embedding as input and employ an unsupervised pre-training procedure on larger abundant data. Incorporating both hierarchical ontology information and pre-training procedure, the end-to-end model

Methods	Jaccard	PR-AUC	F1	# of parameters
LR	0.4075	0.6716	0.5658	-
GRAM	0.4176	0.6638	0.5788	3,763,668
LEAP	0.3921	0.5855	0.5508	1,488,148
RETAIN	0.4456	0.6838	0.6064	2,054,869
GAMENet$^-$	0.4401	0.6672	0.5996	5,518,646
GAMENet	0.4555	0.6854	0.6126	5,518,646
G-BERT$_{G-,P-}$	0.4186	0.6649	0.5796	2,634,145
G-BERT$_{G-}$	0.4299	0.6771	0.5903	2,634,145
G-BERT$_{P-}$	0.4236	0.6704	0.5844	3,034,045
G-BERT	**0.4565**	**0.6960**	**0.6152**	3,034,045

Table 3: Performance on Medication Recommendation Task.

G-BERT has more capacity and achieve comparable results with others.

As for baseline models, LR and Leap are worse than our most basic model (G-BERT$_{G-,P-}$) in terms of most metrics. Comparing G-BERT$_{P-}$ and GRAM, which both used medical ontology information without pre-training, the scores of our G-BERT$_{P-}$ is slightly higher in all metrics. This can demonstrate the validness of using Transformer encoders and the specific prediction layer for medication recommendation. Our final model G-BERT is also better than the attention based model, RETAIN, and the recently published state-of-the-art model, GAMENet. Specifically, even adding the extra information of DDI knowledge and procedure codes, GAMENet still performs worse than G-BERT.

In addition, we visualized the pre-training medical code embeddings of G-BERT$_{G-}$ and G-BERT to show the effectiveness of ontology embedding using online embedding projector [2] shown in (https://raw.githubusercontent.com/jshang123/G-Bert/master/saved/tsne.png/).

6 Conclusion

In this paper we proposed a pre-training model named G-BERT for medical code representation and medication recommendation. To our best knowledge, G-BERT is the first that utilizes language model pre-training techniques in healthcare domain. It adapted BERT to the EHR data and integrated medical ontology information using graph neural networks. By additional pre-training on the EHR from patients who only have one hospital visit which are generally discarded before model training, G-BERT outperforms all baselines in prediction accuracy on medication recommendation task. One direction for the future work is to add more auxiliary and structural tasks to improve the ability of code representaion. Another direction may be to adapt our model to be suitable for even larger datasets with more heterogeneous modalities.

[2] https://projector.tensorflow.org/

References

[Baytas *et al.*, 2017] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Choi *et al.*, 2016] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

[Choi *et al.*, 2017] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: Graph-based attention model for healthcare representation learning. In *SIGKDD*, 2017.

[Choi *et al.*, 2018] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4547–4557. Curran Associates, Inc., 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Erhan *et al.*, 2010] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[Gilmer *et al.*, 2017] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, and G.E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.

[Hinton *et al.*, 2006] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Lipton *et al.*, 2015] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[Luaces *et al.*, 2012] Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313, 2012.

[Ma *et al.*, 2018] Tengfei Ma, Cao Xiao, and Fei Wang. Health-atm: A deep architecture for multifaceted patient health record representation and risk prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 261–269. SIAM, 2018.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, volume 1, pages 2227–2237, 2018.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Ramachandran *et al.*, 2016] Prajit Ramachandran, Peter J Liu, and Quoc V Le. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*, 2016.

[Shang *et al.*, 2019] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. Gamenet: Graph augmented memory networks for recommending medication combination. *AAAI*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 1(2), 2017.

[Xiao *et al.*, 2018a] Cao Xiao, Edward Choi, and Jimeng Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 2018.

[Xiao *et al.*, 2018b] Cao Xiao, Tengfei Ma, Adji B. Dieng, David M. Blei, and Fei Wang. Readmission prediction via deep contextual embedding of clinical concepts. *PLOS ONE*, 13(4):1–15, 04 2018.

[Zhang *et al.*, 2017] Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng Sun. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. In *SIGKDD*, pages 1315–1324, 2017.