Joint Medical Ontology Representation Learning for Healthcare Predictions

Ke Wang[†‡], Ning Chen[†‡] and Ting Chen[†‡*]
[†] Institute for Artificial Intelligence, Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
[‡] Tsinghua-Fuzhou Institute of Digital Technology, Beijing National Research Center
for Information Science and Technology, Tsinghua University, Beijing 100084, China
Email: *tingchen@mail.tsinghua.edu.cn

***Abstract*—Healthcare predictions aim at predicting diseases of the next visit to hospital with historical Electronic Health Records (EHR), which is a key research field in personalized healthcare. Previous research has demonstrated that learning meaningful medical ontology representations within the healthcare prediction model can alleviate the data insufficiency problem and thus is beneficial to this task. There are two main pathways of learning medical ontology representations. The first is through pre-defined knowledge graph such as the ICD tree, and the second is through the co-occurrence of diseases within each visit. Majority of existing works formalize their model under only one pathway, and fail to utilize the mutual benefits between them. To exploit these benefits, we propose JMRL, an end-to-end and accurate model for healthcare predictions with Joint Medical ontology Representation Learning. JMRL not only utilizes the joint information from both knowledge graph and co-occurrence statistics, but also make use of the mutual benefits between them in an advanced way with two explicit feedback strategies. Experimental results on the MIMIC-III dataset demonstrate the superiority of our model over all existing state-of-the-art approaches.**

***Index Terms*—medical ontology representation, healthcare predictions, joint learning**

I. Introduction

Electronic Health Records (EHR) are temporal sequential data of patient medical visits, consisting of diagnosis codes, medications and lab examination results. With the widespread adoption of EHR, there is a rapid growth in the volume and diversity of healthcare data, which motivates the research in applying clinical decision models to increase the quality of healthcare services [1]. Among all the tasks in the healthcare field, predicting patients' next illness with previous records is still a challenging one, and has become a popular research field for decades [2], [3].

The task of healthcare predictions is to predict a patient's next diagnosis codes with diagnosis codes of previous visits to hospital. The temporal feature of this task makes Recurrent Neural Network (RNN) a natural choice to model the sequential relations among different visits. For instance, Dipole [3] applies bidirectional RNN with different attention strategies. RETAIN [4] utilizes a reverse time attention mechanism to calculate the importance of each diagnosis code that has appeared in former visits for current prediction.

One key problem of this task is how to model the highly discrete diagnosis codes, which are the input to the model. Besides, a well-learned representation of medical ontologies will greatly benefit downstream tasks such as automatic diagnosis [5], mortality prediction [6] and healthcare question answering [7].

Traditional methods use one-hot coding [8], which is highly sparse and requires huge amounts of data for model training, leading to data insufficiency problems. Human designed feature representations are adopted in [9], but this method suffers from scalability problems. Med2Vec [2] is the first to utilize a code embedding matrix and embeds each medical code to a non-negative real-valued high-dimensional vector, but this method still regards each medical code as totally independent. To utilize the intrinsic relationship between medical codes, GRAM [10] introduces the hierarchical structure of the ICD-9 tree as an external knowledge graph, and applies graph-based attention mechanism to learn robust medical code embeddings with ancestors. KAME [11] extends GRAM [10] by directly exploiting medical knowledge in the whole prediction process. MMORE [12] enables each non-leaf medical code in the ICD-9 tree to possess multiple ontological representations, which adds to the diversity of code expressiveness.

In addition to the utilization of external knowledge graph such as the ICD-9 tree, the co-occurrence statistics of medical codes are also explored to mine the relationship between medical codes. The theoretical foundation behind co-occurrence statistics is that medical codes that often co-occur tend to be close in the embedding space. Skip-gram [13] is adopted by [2] to utilize the co-occurrence information within each visit to hospital. CBOW [13] and time-aware attention mechanism are employed in [14], where each medical code is assigned a weight distribution within a time period.

Although previous works have made great contributions to medical ontology representation learning, they focus solely on either knowledge graph or co-occurrence statistics, or model them jointly in an inefficient or inappropriate manner. GRAM [10] utilizes GloVe [15] to initialize the embedding matrix for graph-based attention. Med2Vec [2] imposes topological and co-occurring regulations on the same vector space, which is highly probable to raise conflicts during training.

To solve these problems and explore the mutual benefits be-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

tween knowledge graph and co-occurrence statistics, we propose JMRL, an end-to-end and accurate model for healthcare predictions with Joint Medical ontology Representation Learning. First, our model adopts two separate embedding spaces for topological feature and co-occurrence feature, which prevents the potential conflicts during the training process. Second, we incorporate our model with two explicit feedback strategies between the two feature spaces to bond them together. As a result of the feedback strategies, the topological embedding space can benefit from the improvement of the co-occurrence embedding space and vice versa. In addition, our method adopts an attentive Gated Recurrent Unit (GRU) to better aggregate the information from previous visits and give more accurate predictions.

We evaluate our model on the public available MIMIC-III [16] dataset, and compare it with a number of competitive baselines. Experimental results indicate that our method can not only increase the prediction accuracy, but also improve the quality of medical ontology representations. We also conduct an ablation analysis to further verify the internal mechanisms of our proposed model.

II. Proposed Model

In this section, we first introduce and give the definitions related to EHR data and medical ontologies. Then, we present the detailed descriptions of our proposed model JMRL, including co-occurrence embedding, knowledge graph embedding and attentive prediction module. The two explicit feedback strategies are explained in knowledge graph embedding and co-occurrence embedding separately. The initialization of embedding matrices is introduced in the end. The overall structure of JMRL is shown in Fig. 1.

A. Basic Notations

We denote the entire set of diagnosis codes from the EHR dataset as $c_1, c_2, \cdots, c_{|\mathcal{C}|} \in \mathcal{C}$, and $|\mathcal{C}|$ is the total number of unique medical codes. In our experiment, we only consider diagnosis codes as medical codes for consistence with previous works such as [2], [10] and [11]. The ICD-9 tree, which is a directed acyclic graph (DAG), contains the hierarchy of various medical concepts with the parent-child relationship, and the specificity of medical concepts increases with depth. Only the leaf node in the ICD-9 tree represents a medical code in $\mathcal{C}$. We define the non-leaf nodes as $\{c_{|\mathcal{C}|+1}, c_{|\mathcal{C}|+2}, \cdots, c_{|\mathcal{C}|+|\mathcal{C}'|}\}$, where $|\mathcal{C}'|$ is the number of non-leaf nodes in the ICD-9 tree.

The EHR data is a temporal sequence of patients' visits to hospital. For the p-th patient with $T^{(p)}$ visits to hospital, his/her EHR can be represented by a sequence of visits $P_1, P_2, \cdots, P_{T^{(p)}}$. Each visit P_t is a subset of medical code set $\mathcal{C}$, and it can be represented as a binary vector $x_t \in \{0,1\}^{|\mathcal{C}|}$, where the i-th element is 1 only if P_t contains the medical code c_i.

Here we define two basic embedding matrices $E \in \mathbb{R}^{\left(|\mathcal{C}|+|\mathcal{C}'|\right)\times k_E}$ and $V \in \mathbb{R}^{\left(|\mathcal{C}|+|\mathcal{C}'|\right)\times k_V}$ to be the medical ontology representations to be learned from the DAG and co-occurrence statistics respectively, where k_V and k_E are the dimensions of embeddings. We use e_i and v_i to represent the basic knowledge graph embedding and basic co-occurrence embedding of medical code c_i.

For simplicity, we describe our model in the following parts for one patient with T visits to hospital.

B. Co-occurrence Embedding

The foundation of training co-occurrence embedding is similar to learning word embeddings, where frequently co-occurring medical codes should be close in the embedding space. Previous methods simply optimize on each pair of medical codes that co-occur within the same visit. They fail to notice a fact that it is normal for people to carry different diseases at the same time though they are not closely related from the medical view, e.g. bronchitis and diarrhea. As a result, simply optimizing over all pairs of medical codes within the same visit is not accurate.

To solve this problem, we improve upon previous works by adding weights to different pairs of medical codes within the same visit. This modification also contains **the first feedback strategy** in our model.

The co-occurrence loss for training is as follows. To introduce **the feedback from knowledge graph embedding matrix** E, we use the knowledge graph embedding e_i and e_j to calculate the weight of code pair (c_i, c_j).

$$\mathcal{L}_{co} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i:c_i\in P_t}\sum_{j:c_j\in P_t, j\neq i}\beta_{ij}\cdot(1-p(c_i|c_j)) \\ \text{where } \beta_{ij} = \sigma\left(e_i^T e_j\right),\ p(c_i|c_j) = \sigma\left(v_i^T v_j\right) \tag{1}$$

where σ denotes sigmoid activation function. The weight β_{ij} depicts the degree of relevance between medical code c_i and c_j, and it bonds the two embedding spaces together. If the knowledge graph embedding is optimized during the training process, it will guide the co-occurrence loss $\mathcal{L}_{co}$ to be more reasonable and accurate.

The co-occurrence loss $\mathcal{L}_{co}$ only considers leaf nodes in the DAG. In order to propagate the information from leaf nodes to non-leaf nodes, we utilize the parent-children relationship in the DAG with the self-attention mechanism [18].

For each non-leaf node c_i, its final co-occurrence representation $v_i^{'}$ is the weighted summation of its children and the feature vector of itself v_j. The propagation process is conducted from nodes with larger depth to nodes with smaller depth to ensure proper calculation order:

$$v_i^{'} = \sum_{j\in\mathcal{N}(i)}\gamma_j v_j + v_i \\ \text{where } \gamma_j = \text{Softmax}\left(\frac{(v_j W^Q)(v_j W^K)^T}{\sqrt{d_k}}\right)(v_j W^V) \tag{2}$$

where $W^K \in \mathbb{R}^{k_V\times d_k}, W^Q \in \mathbb{R}^{k_V\times d_k}$, $W^V \in \mathbb{R}^{k_V\times k_V}$ are parameter matrices and d_k is the dimension of queries and keys. $\mathcal{N}(i)$ represents the set of neighboring nodes of node

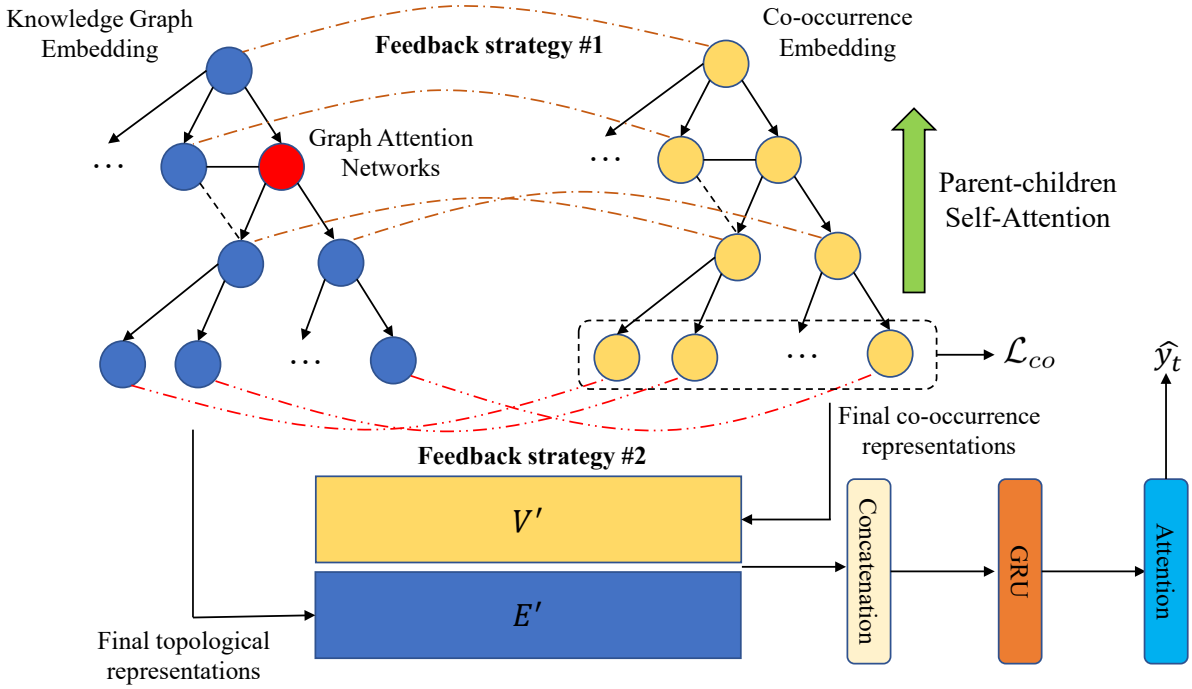

Fig. 1. Overview of the proposed JMRL model. The left ICD-9 tree indicates knowledge-graph embedding, and the right part represents co-occurrence embedding. The brown dotted line on the top represents the #1 feedback from co-occurrence embedding to knowledge graph embedding, and the red dotted line on the bottom represents the #2 feedback from knowledge graph embedding to co-occurrence embedding. The red node in the knowledge graph embedding stands for the node that graph attention networks are working on. It is noticeable that though we omit many nodes for simplicity, the ICD-9 tree is not a binary tree. The black dashed line in the ICD-9 tree indicates the relationship extracted from KnowLife [17].

i. After this process, we get the final representation of co-occurrence feature $V^{'} \in \mathbb{R}^{\left(|\mathcal{C}|+|\mathcal{C}^{'}|\right)\times k_V}$.

C. Knowledge Graph Embedding

Previous works generally utilize the parent-children relationship and use the basic embeddings of ancestor nodes to generate a robust representation of the leaf node, which stands for a medical code. However, considering only the parent-children relationship will ignore the potential connections with adjacent non-ancestor nodes. In our model, we use Graph Neural Networks (GNN) to better capture the relationships among nodes in the knowledge graph.

We formalize the final representation of nodes' topological feature as the output of multiple Graph Attention (GAT) [19] layers. GAT can integrate the information from neighboring nodes with respect to their attention weights. We make use of this property and modify GAT's structure to establish **the second feedback strategy**.

As we adopt multiple GAT layers, we demonstrate the calculation process at an arbitrary layer l for simplicity. For node c_i, we first calculate its attention weight with neighboring node $c_j \in \mathcal{N}(i)$ as follows:

$$\alpha_{ij}^{(l)} = \frac{exp\left(e_{ij}^{(l)}\right)}{\sum_{k\in\mathcal{N}(i)} exp\left(e_{ik}^{(l)}\right)} \tag{3}$$

where $e_{ij}^{(l)}$ is the pair-wise un-normalized attention score between node c_i and c_j. To introduce **the feedback from co-occurrence embedding matrix** V, we compute $e_{ij}^{(l)}$ via the following feed-forward network:

$$\begin{aligned} e_{ij}^{(l)} &= \text{LeakyReLU}\left(W_e^{(l)}\left(z_i^{(l)}||z_j^{(l)}\right)\right) \\ \text{where } z_k^{(l)} &= W_z\left(h_k^{(l)}||v_k^{'}\right) \quad k \in \{i,j\} \end{aligned} \tag{4}$$

where W_e and W_z are parameter matrices, $||$ denotes concatenation, $v_k^{'}$ is the final representation of co-occurrence embedding for node c_k, and $h_i^{(l)}$ is the output vector for node c_i at layer $l-1$. We define $h_i^{(1)} = e_i$. The co-occurrence embedding $v_k^{'}$ is fed into the knowledge graph embedding as part of the input, which bonds the two embedding spaces together. If the co-occurrence embedding matrix V is optimized during the training process, it will guide the GAT layer to assign attention weights more precisely and efficiently.

Next, we aggregate the information from neighboring nodes to get layer l's output $h_i^{(l+1)}$. We add shortcut connections between adjacent GAT layers to prevent the information smoothing problem:

$$h_i^{(l+1)} = \sum_{j\in\mathcal{N}_i} \alpha_{ij}^{(l)} h_j^{(l)} + h_i^{(l)} \tag{5}$$

After multiple GAT layers, we get the final topological

representations $E^{'} \in \mathbb{R}^{\left(|\mathcal{C}|+|\mathcal{C}^{'}|\right)\times k_E}$.

D. Attentive Prediction Module

After we get $E^{'}$ and $V^{'}$, the final representations of both knowledge graph embedding and co-occurrence embedding, we concatenate them row-wisely as the final representations of medical ontologies, i.e., $S = E^{'}||V^{'}$. Given a patient's visits to hospital $P_1, P_2, \cdots, P_T$ with their vector representations $x_1, x_2, \cdots, x_T$, we first transfer them with the final medical ontologies representations, i.e. $a_t = S^T x_t$.

Next, we apply a GRU layer to mine the temporal information between different visits:

$$h_1, h_2, \cdots, h_T = GRU(a_1, a_2, \cdots, a_T) \tag{6}$$

The dimension of the hidden state h_i is d_h. When predicting the diagnosis codes of the t^{th} visit with the previous $(t-1)$ visits, we aggregate the previous information attentively as follows:

$$\begin{aligned} \overline{a}_t &= \sum_{i=1}^{t-1} \rho_i * h_i \\ \text{where } \rho_i &= \frac{exp\left(W_a h_i\right)}{\sum_{j=1}^{t-1} exp\left(W_a h_j\right)} \end{aligned} \tag{7}$$

where W_a is a multi-layer perceptron. The final prediction $\hat{y}_t$ is calculated via another linear layer:

$$\hat{y}_t = \text{Sigmoid}\left(W_o \overline{a}_t + b_o\right) \tag{8}$$

where $W_o \in \mathbb{R}^{|\mathcal{C}|\times d_h}$ and bias $b_o \in \mathbb{R}^{|\mathcal{C}|}$ are learnable parameters.

We optimize over predicting all visits except the first one, and the objective function for optimization is as follows:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{pre} + \lambda \mathcal{L}_{co} \text{ where} \\ \mathcal{L}_{pre} &= \frac{-1}{T-1} \sum_{t=2}^{T} \left(y_t^T \log(\hat{y}_t) + (1-y_t)^T \log(1-\hat{y}_t)\right) \end{aligned} \tag{9}$$

where $\lambda \in (0, 1)$ is a hyper-parameter for adjustment during experiments. The whole training procedure is described in Algorithm 1. For simplicity, we consider only one patient in a mini-batch of training data in the algorithm.

E. Embedding Initialization

A proper initialization of the embedding matrices E and V can greatly benefit the training process. For basic knowledge graph embedding matrix E, we use the average vector of pretrained fastText [20] embeddings of each medical code's description from the CCS-multi-level diagnosis hierarchy[1] to initialize it. For initializing the basic co-occurrence embedding matrix V, we follow the original settings described in [10] with GloVe.

Algorithm 1 JMRL Training Algorithm

Input: Training dataset with N patients, and each patient with T_i visits to hospital. The DAG structure of ICD-9 hierarchy (for GAT layer).

```
Initialize basic knowledge graph embedding E, basic co-occurrence embedding V and other model parameters.
repeat
    X ← random patient from the training dataset with T visits
    Calculate the co-occurrence loss L_co with Eq. (1)
    Calculate the final representations of co-occurrence feature V' with Eq. (2)
    Calculate the final representations of topological feature E' with Eq. (3)-(5)
    Initialize the total prediction loss L_pre = 0
    for visit V_t, t ∈ [1, T-1] in X do
        Calculate prediction ŷ_{t+1} with E' and V' in Attentive Prediction Module with Eq. (6)-(8)
        Calculate prediction loss for V_{t+1} and add it to L_pre
    end for
    Calculate the total loss L with co-occurrence loss L_co and next illness prediction loss L_pre
    Update model parameters with loss L and optimizer
until model convergence
```

III. Experiments

In this section, we first introduce the MIMIC-III [16] dataset that we conduct experiments on, including dataset statistics and data pre-processing methods. Then, we compare our proposed model JMRL with several state-of-the-art baselines in terms of the prediction accuracy for diagnosis prediction of the next visit to hospital. The ability of model to handle data insufficiency situations are also explored. Besides, we conduct ablation analysis with several modifications to our model to verify the effects of the two explicit feedback strategies designed in our model.

A. Dataset and Pre-processing

MIMIC-III [16] is a large and publicly-available dataset containing more than $60,000$ de-identified intensive care unit admission records. In our experiment, we aim to learn meaningful representations of diagnosis codes and improve the accuracy of healthcare prediction simultaneously.

During the data pre-processing procedure, we filter out patients with less than two visits to the hospital. After data pre-processing, we extract 7537 patients with an average of 2.65 visits to hospital. The average number of diagnosis codes in each visit is 12.90. We randomly divide the dataset into the training, validation and test dataset by a ratio of 0.8:0.1:0.1, and use the validation dataset to tune the hyper-parameters.

B. Baseline Approaches

To validate the predictive performance of our proposed model, we compare it with the following models:

[1]https://bioportal.bioontology.org/ontologies/ICD9CM

TABLE I
RESULTS OF VISIT-LEVEL PRECISION@K WITH TWO DIFFERENT VALUES OF k AND VARYING SIZE OF THE TRAINING DATASET FROM 20% TO 100%. THE RATIO INDICATES THE SIZE OF THE UTILIZED TRAINING DATASET.

Model	k=10					k=20				
	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
RETAIN [4]	0.5834	0.6017	0.6342	0.6591	0.6702	0.6012	0.6883	0.7221	0.7402	0.7526
Dipole [3]	0.5809	0.6102	0.6219	0.6614	0.6764	0.5876	0.6988	0.7337	0.7347	0.7507
GRAM [10]	0.6271	0.6674	0.6902	0.6951	0.7078	0.7051	0.7449	0.7617	0.7662	0.7716
KAME [11]	0.6113	0.6772	0.6941	0.7001	0.7119	0.6882	0.7329	0.7631	0.7704	0.7731
MMORE [12]	0.6205	0.6613	0.7011	0.7052	0.7093	0.6957	0.7413	0.7682	0.7704	0.7721
JMRL	0.6362	0.6855	**0.7118**	0.7174	0.7225	0.7182	0.7591	0.7709	**0.7816**	**0.7839**
JMRL+	**0.6481**	**0.6903**	0.7095	**0.7181**	**0.7233**	**0.7229**	**0.7633**	**0.7717**	0.7799	0.7821

RETAIN [4]. RETAIN is an interpretable prediction model with a two-level attention mechanism.

Dipole [3]. Dipole utilizes bidirectional RNN to predict next illness with three different attention mechanisms.

GRAM [10]. GRAM is the first model to utilize the hierarchy of external medical knowledge, and it employs graph-based attention to learn robust representations of medical codes.

KAME [11]. KAME extends GRAM by directly exploiting medical knowledge in the whole prediction process. The direct application of ancestor nodes can increase prediction accuracy and interpretability.

MMORE [12]. MMORE extends GRAM by allowing non-leaf nodes in the DAG to possess multiple semantic meanings, which enables nodes with the same ancestor to possess relatively different meanings.

To verify the effect of initialization, we test our model both with and without initialization. We use JMRL to denote our model with no initialization and JMRL+ to denote our model with initialization.

C. Experiment Settings

We adopt the CCS-multi-level diagnosis hierarchy[1] to build the knowledge graph. In order to introduce more relationships among medical ontologies in the DAG, we extract expert medical knowledge from KnowLife [17] such as the relationship of *cause* and *is caused by*. Besides, we add edges between nodes that have the same nearest ancestor to encourage more information transmission between nearby nodes.

All experiments are implemented with the PyTorch [21] framework. For all models, we use Adam [22] optimizer with an initial learning rate of 0.001 and mini-batch size of 64. The dimensions of both knowledge graph embedding and co-occurrence embedding are set to 300. The number of GAT layers is set to 4 and λ is set to 0.2 after multiple experiments. For baseline models, we strictly follow the experiment settings provided in the original paper.

To improve the training convergence and predictive performance, we follow the settings of previous works [10]–[12] and group the label set into 169 different groups with the CCS-multi-level diagnosis hierarchy[1] as the real label set for prediction. The 169 groups can still preserve the sufficient granularity for each diagnosis.

D. Evaluation Metrics

The task of healthcare prediction is similar to the task of personalized recommendations [23]. Therefore, we adopt two commonly-used metrics to fully evaluate the performance of our proposed model, namely *visit-level precision@k* and *code-level accuracy@k*, where k is a parameter.

Visit-level precision@k is used to evaluate coarse-grained model performance from the visit level. It is defined for each visit as follows:

$$\text{precision@k} = \frac{\#\text{ of true positives in top } k}{\min(k, |y_t|)} \tag{10}$$

Code-level accuracy@k is adopted to evaluate the fine-grained model performance at the code level. We calculate accuracy@k for each code in the real label set for prediction as follows. Given a visit V_t, we get 1 if the target diagnosis code is in the top-k predictions and 0 otherwise. We average across all visits in the dataset to get the value of accuracy@k for each code.

We take the average value over the test dataset as the final value of precision@k, and take the average value over the specified label space according to code frequency as the final value of accuracy@k.

E. Results and Discussions

In the experiment, we want to explore two aspects of our model. The first is whether our model can provide accurate predictions of the next visit illness with different values of k. The second is whether our model can surpass baseline models with insufficient data such as 20% or 40% of the original training dataset. For code-level accuracy@k, as the size of the true label set is large (169), we divide the true label set into 5 categories by the percentile of their frequencies in a non-decreasing order in the training dataset and calculate the average value of each category to better demonstrate the model performance.

Table I shows the result of next visit illness prediction with different values of k and varying sizes of the training dataset. To mimic the situation of data insufficiency, we test all models

TABLE II
RESULTS OF CODE-LEVEL ACCURACY@K WITH TWO DIFFERENT VALUES OF k. THE RATIO INDICATES THE PERCENTILE OF CODE FREQUENCIES IN THE TRAINING DATASET. FOR EXAMPLE, THE COLUMN OF $0-20\%$ REFERS TO THE AVERAGE VALUE OF CODE-LEVEL ACCURACY@K AMONG CODES WHOSE PERCENTILE OF FREQUENCIES IN NON-DECREASING ORDER ARE WITHIN THE RANGE OF 0% TO 20%.

Model	k=10					k=20				
	0-20%	20-40%	40-60%	60-80%	80-100%	0-20%	20-40%	40-60%	60-80%	80-100%
RETAIN [4]	0.0014	0.0089	0.1125	0.1715	0.5663	0.0032	0.0278	0.1526	0.4398	0.7862
Dipole [3]	0.0041	0.0143	0.1016	0.1823	0.5765	0.0041	0.0327	0.1496	0.4366	0.7824
GRAM [10]	0.0069	0.0548	0.1911	0.3289	0.6158	0.0128	0.1095	0.3607	0.6337	0.7850
KAME [3]	0.0052	0.0482	0.2117	0.3432	0.6289	0.0098	0.1113	0.3552	0.6449	0.7901
MMORE [12]	0.0041	0.0511	0.1852	0.3522	0.6201	0.0107	0.1225	0.3662	0.6452	0.7861
JMRL	0.0104	0.0725	**0.2352**	**0.3617**	0.6388	0.0286	0.1428	**0.3789**	0.6501	**0.8007**
JMRL+	**0.0152**	**0.0787**	0.2331	0.3591	**0.6402**	**0.0355**	**0.1501**	0.3722	**0.6513**	0.7986

with varying sizes of the original training dataset. From the results we can observe the following facts. The first is that our proposed model JMRL can exceed all baselines with a clear margin under different settings of k. Second, JMRL can better adapt to the environment of data insufficiency compared with other baselines. This is natural because JMRL can absorb information from both knowledge graph and co-occurrence statistics. The feedback strategies also enable the network to learn faster with limited data. Another noticeable point is that a good initialization can greatly benefit the training procedure of our model when data is extremely limited such as only 20% of the original training dataset. However, the promotion from initialization will gradually disappear as the volume of training data increases. The last thing is that GRAM, KAME and MMORE share similar prediction performance, which indicates that digging deeper solely in the knowledge graph embedding will not bring about apparent improvements.

Table II shows the result of code-level accuracy@k with different values of k. We can see that our model performs better on infrequent codes compared with baselines. The reason for the relatively poor performance of infrequent nodes is that they appear too few times in the training dataset. For JMRL, the joint utilization of knowledge graph constraints and co-occurring constraints enables the model to assign similar embeddings to codes with strong connections regardless of their infrequency, thus improving the chances that these infrequent codes are ranked within the top-k predictions.

F. Ablation Analysis

To verify the effects of the two feedback strategies that we propose in our model, we modify our model and conduct multiple experiments as follows:

1) **JMRL**: We keep the original settings of our model unchanged, where both feedback strategies are kept.
2) **JMRL-1**: We remove the first feedback strategy from our model, which is the feedback from co-occurrence embedding to knowledge graph embedding.
3) **JMRL-2**: We remove the second feedback strategy from our model, which is the feedback from knowledge graph embedding to co-occurrence embedding.
4) **JMRL-1-2**: We remove both feedback strategies from our model. The two embedding spaces are totally independent in this setting.

TABLE III
RESULTS OF VISIT-LEVEL PRECISION@K IN ABLATION ANALYSIS.

Model	Ratio of Training Dataset Size				
	20%	40%	60%	80%	100%
JMRL	0.7182	0.7591	0.7709	0.7816	0.7839
JMRL-1	0.7039	0.7512	0.7632	0.7723	0.7781
JMRL-2	0.6977	0.7472	0.7656	0.7698	0.7739
JMRL-1-2	0.6813	0.7301	0.7574	0.7615	0.7652
GRAM [10]	0.7051	0.7449	0.7617	0.7662	0.7716

For simplicity, we set k to be 20 and only vary the size of the training dataset. The results of the ablation analysis are shown in Table III.

From the results we can see that the two feedback strategies are truly important in our proposed model. Removing either the first or the second feedback strategy will result in obvious decrease in prediction accuracy. However, it is noticeable that with only one feedback strategy, our model can still maintain some advantages over baseline models. If we remove both feedback strategies at the same time, the performance will drop below that of GRAM or KAME, which indicates that simply applying knowledge graph embedding and co-occurrence embedding at the same time but independently will not lead to improvements of model performance. Only by bonding the two embedding spaces together with explicit feedback strategies between each other, as in our proposed model, can the prediction accuracy be truly improved.

IV. CONCLUSION

In this paper, we propose JMRL, an end2end and accurate model for healthcare predictions with Joint Medical ontology Representation Learning. To make full use of the information from healthcare data, we adopt both knowledge graph embedding and co-occurrence embedding. Besides, we design two explicit feedback strategies between the two embedding

spaces to explore the mutual benefits between them. Experimental results on the public MIMIC-III dataset demonstrate that compared with the state-of-the-art baselines, our model can improve the accuracy of next illness prediction, adapt better to the data insufficiency environment and perform better on infrequent codes. Besides, we conduct ablation analysis to further verify the effects of two feedback strategies that we design in our model. The results from ablation analysis indicate that the two feedback strategies in our model are essential and removing either of them will lead to performance decrease.

In the future, we will explore how to better model the co-occurrence property, and take the time interval between adjacent visits into consideration.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (grants 61872218, 61721003, 61673241 and 61906105), National Key R&D Program of China (2019YFB1404804), Tsinghua-Fuzhou Institute of Digital Technology, Beijing National Research Center for Information Science and Technology (BNRist), and Tsinghua University-Peking Union Medical College Hospital Initiative Scientific Research Program. The funders had no roles in study design, data collection and analysis, the decision to publish, and preparation of the manuscript. Besides, the author would like to thank Yudong Chen and Nianlong Song for their sincere and helpful suggestions on paper writing.

REFERENCES

[1] Z. C. Lipton, D. C. Kale, C. Elkan, and R. C. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[2] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun, "Multi-layer representation learning for medical concepts," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1495–1504.

[3] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2017, pp. 1903–1911.

[4] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, 2016, pp. 3504–3512.

[5] A. Rios and R. Kavuluru, "Few-shot and zero-shot multi-label learning for structured label spaces," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 3132–3142.

[6] Y. Sha and M. D. Wang, "Interpretable predictions of clinical outcomes with an attention-based recurrent neural network," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2017, pp. 233–240.

[7] J. E. Daniel, W. Brink, R. Eloff, and C. Copley, "Towards automating healthcare question answering in a noisy multilingual low-resource setting," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 948–953.

[8] R. Chen, H. Su, M. Khalilia, S. Lin, Y. Peng, T. Davis, D. A. Hirsh, E. Searles, J. Tejedor-Sojo, M. Thompson *et al.*, "Cloud-based predictive modeling system and its application to asthma readmission prediction," in *AMIA Annual Symposium Proceedings*, vol. 2015. American Medical Informatics Association, 2015, p. 406.

[9] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits, "Unfolding physiological state: Mortality modelling in intensive care units," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 75–84.

[10] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 787–795.

[11] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 743–752.

[12] L. Song, C. W. Cheong, K. Yin, W. K. Cheung, and B. CM, "Medical concept embedding with multiple ontological representations," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 4613–4619.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[14] X. Cai, J. Gao, K. Y. Ngiam, B. C. Ooi, Y. Zhang, and X. Yuan, "Medical concept embedding with time-aware attention," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2018, pp. 3984–3990.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1532–1543.

[16] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[17] P. Ernst, A. Siu, and G. Weikum, "Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences," *BMC bioinformatics*, vol. 16, no. 1, p. 157, 2015.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *CoRR*, vol. abs/1710.10903, 2017. [Online]. Available: http://arxiv.org/abs/1710.10903

[20] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, 2017, pp. 427–431.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] R. Catherine and W. Cohen, "Personalized recommendations using knowledge graphs: A probabilistic logic programming approach," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 325–332.