A General Framework for Diagnosis Prediction via Incorporating Medical Code Descriptions

Fenglong Ma[1], Yaqing Wang[1], Houping Xiao[2], Ye Yuan[3], Radha Chitta[4], Jing Zhou[5], Jing Gao[1]
[1]University at Buffalo, [2]Georgia State University, [3]Beijing University of Technology, [4]Kira Systems, [5]eHealth Inc.
[1]{fenglong, yaqingwa, jing}@buffalo.edu, [2]hxiao@gsu.edu,
[3]yuanye91@emails.bjut.edu.cn, [4]radha.cr@gmail.com, [5]jing.zhou@ehealth.com

***Abstract*—Diagnosis prediction aims to predict the future health status of patients according to their historical visit records, which is an important yet challenging task in healthcare informatics. Existing diagnosis prediction approaches mainly employ recurrent neural networks (RNNs) with attention mechanisms to make predictions. However, these approaches ignore the importance of code descriptions, i.e., the medical definitions of diagnosis codes. We believe that taking diagnosis code descriptions into account can help the state-of-the-art models not only to learn meaningful code representations, but also to improve the predictive performance. Thus, in this paper, we propose a simple, but general diagnosis prediction framework, which includes two basic components: diagnosis code embedding and predictive model. To learn the interpretable code embeddings, we apply convolutional neural networks (CNNs) to model medical descriptions of diagnosis codes extracted from online medical websites. The learned medical embedding matrix is used to embed the input visits into vector representations, which are fed into the predictive models. Any existing diagnosis prediction approach (referred to as the base model) can be cast into the proposed framework as the predictive model (called the enhanced model). We conduct experiments on two real medical datasets: the MIMIC-III dataset and the Heart Failure claim dataset. Experimental results show that the enhanced diagnosis prediction approaches significantly improve the prediction performance.**

I. Introduction

Due to the immense accumulation of Electronic Healthcare Records (EHR), it is possible to directly predict patients' future health status according to their historical visit records [1]–[14]. Especially, *diagnosis prediction*, which aims to predict the diagnosis information of patients in the following visits, attracts considerable attention from both healthcare providers and researchers. The key challenge of diagnosis prediction task is how to design an accurate and robust predictive model to handle the temporal, high dimensional and noisy EHR data.

Recently, recurrent neural networks (RNN) based diagnosis prediction models [2], [7], [8] have been broadly applied to tackle these challenges. RETAIN [8] uses two recurrent neural networks with attention mechanisms to model the reverse time ordered EHR sequences. Dipole [2] enhances the prediction accuracy by employing a bidirectional recurrent neural network (BRNN) with different attention mechanisms. The aforementioned models typically require large amounts of EHR training data to guarantee the predictive performance. However, there always exist medical codes of rare diseases, which infrequently appear in the EHR data. To overcome this issue, GRAM [7] has been proposed, which learns medical code representations by exploiting medical ontology information and the graph-based attention mechanism. For the rare medical codes, GRAM can alleviate the difficulties of learning their embeddings by considering their ancestors' embeddings to guarantee the predictive performance. However, the performance of GRAM heavily depends on the choice of medical ontology. Thus, without specific input constraints, how to learn robust embeddings for medical codes is still the major challenge for accurate diagnosis prediction.

To resolve this challenge, we consider the "nature" of diagnosis codes, i.e., their medical descriptions. Actually, each diagnosis code has a formal description, which can be easily obtained from the Internet, such as Wikipedia or online medical websites. For example, from ICD9Data.com, the description of diagnosis code "428.32" is "*Chronic diastolic heart failure*", and "*Rheumatic heart failure (congestive)*" is the description of diagnosis code "398.91". Without considering the medical meanings of diagnosis codes, they are treated as two independent diseases in the EHR dataset. However, they both describe the same disease, i.e., "heart failure". Thus, we strongly believe that **incorporating the descriptions of diagnosis codes** should help the predictive models to improve the prediction accuracy.

The other benefit of incorporating diagnosis code descriptions is that it enables us to design a **general diagnosis prediction framework**. The input data of all the existing diagnosis prediction approaches are the same, i.e., a sequence of time-ordered visits, and each visit consists of some diagnosis codes. Thus, all the existing approaches, including, but not limited to RETAIN, Dipole and GRAM, can be extended to incorporate the descriptions of diagnosis codes to further improve their predictive performance.

In this paper, we propose a novel framework for diagnosis prediction task. It should be noted that all of the state-of-the-art diagnosis prediction approaches (referred to as *base models*) can be cast into the proposed framework. These base models enhanced by the proposed framework are thus called *enhanced models*. Specifically, the proposed framework consists of two components: diagnosis code embedding and predictive model. The diagnosis code embedding component aims to learn the medical representations of diagnosis codes according to their descriptions. In particular, for each word in the description, we obtain the pretrained vector representation from fastText [15]. Then the concatenation of all the words

978-1-5386-5488-0/18/$31.00 ©2018 IEEE

in each diagnosis code description is fed into a convolutional neural network (CNN) to generate the medical embeddings. Based on the learned medical embeddings of diagnosis codes, the predictive model component makes prediction. It first embeds the input visit information into a visit-level vector representation with the code embeddings, and then feeds this vector into the predictive model, which can be any existing diagnosis prediction approach.

II. Diagnosis Prediction with Code Descriptions

A. Notations

We denote all the unique diagnosis codes from the EHR data as a code set $\mathcal{C} = \{c_1, c_2, \cdots, c_{|\mathcal{C}|}\}$, where $|\mathcal{C}|$ is the number of diagnosis codes. Let $|\mathcal{P}|$ denote the number of patients in the EHR data. For the p-th patient who has T visit records, the visiting information of this patient can be represented by a sequence of visits $\mathcal{V}^{(p)} = \{V_1^{(p)}, V_2^{(p)}, \cdots, V_T^{(p)}\}$. Each visit $V_t^{(p)}$ consists of multiple diagnosis codes, i.e., $V_t^{(p)} \subseteq \mathcal{C}$, which is denoted by a binary vector $\mathbf{x}_t^{(p)} \in \{0,1\}^{|\mathcal{C}|}$. The i-th element of $\mathbf{x}_t^{(p)}$ is 1 if $V_t^{(p)}$ contains the diagnosis code c_i. For simplicity, we drop the superscript (p) when it is unambiguous.

Each diagnosis code c_i has a formal medical description, which can be obtained from Wikipedia[1] or ICD9Data.com[2]. We denote all the unique words which are used to describe all the diagnosis codes as $\mathcal{W} = \{w_1, w_2, \cdots, w_{|\mathcal{W}|}\}$, and $c_i' \subseteq \mathcal{W}$ as the description of c_i, where $|\mathcal{W}|$ is the number of unique words. With the aforementioned notations, the inputs of the proposed framework are the set of code descriptions $\{c_1', c_2', \cdots, c_{|\mathcal{C}|}'\}$ and the set of time-ordered sequences of patient visits $\{\mathbf{x}_1^{(p)}, \mathbf{x}_2^{(p)}, \cdots, \mathbf{x}_{T-1}^{(p)}\}_{p=1}^{|\mathcal{P}|}$. For each timestep t, we aim to predict the information of the $(t+1)$-th visit. Thus, the outputs are $\{\mathbf{x}_2^{(p)}, \mathbf{x}_3^{(p)}, \cdots, \mathbf{x}_T^{(p)}\}_{p=1}^{|\mathcal{P}|}$.

B. Preliminaries

In this subsection, we first introduce the commonly used techniques for modeling patients' visits, and then list all the state-of-the-art diagnosis prediction approaches.

Fully Connected Layer

Deep learning based models are commonly used to model patients' visits. Among existing models, fully connected layer (FC) is the simplest approach, which is defined as follows:

$$\mathbf{h}_t = \mathbf{W}_c \mathbf{v}_t + \mathbf{b}_c, \tag{1}$$

where $\mathbf{v}_t \in \mathbb{R}^d$ is the input data, d is the input dimensionality, $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}| \times d}$ and $\mathbf{b}_c \in \mathbb{R}^{|\mathcal{C}|}$ are the learnable parameters.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) have been shown to be effective in modeling healthcare data [2], [7], [8], [12]. In this paper, GRU is used to adaptively capture dependencies among patient visit information. For simplicity, the GRU can be represented by

$$\mathbf{h}_t = \text{GRU}(\mathbf{v}_t; \Omega),$$

where Ω denotes all the parameters of GRU.

Attention Mechanisms

Attention mechanisms aim to distinguish the importance of different input data, and attention-based neural networks have been successfully used in diagnosis prediction task, including location-based attention [2], [8], general attention [2], concatenation-based attention [2], and graph-based attention [7]. In the following, we introduce two commonly used attention mechanisms: location-based and graph-based attention.

• *Location-based Attention*. Location-based attention mechanism [2], [8] is to calculate the attention score for each visit, which solely depends on the current hidden state $\mathbf{h}_i \in \mathbb{R}^g$ $(1 \leq i \leq t)$ as follows:

$$\alpha_i = \mathbf{W}_\alpha^\top \mathbf{h}_i + b_\alpha, \tag{2}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^g$ and $b_\alpha \in \mathbb{R}$ are the parameters to be learned. According to Eq. (2), we can obtain an attention weight vector $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_t]$ for the t visits. Then the softmax function is used to normalize α. Finally, we can obtain the context vector $\mathbf{c}_t$ according to the attention weight vector α and the hidden states from $\mathbf{h}_1$ to $\mathbf{h}_t$ as follows:

$$\mathbf{c}_t = \sum_{i=1}^{t} \alpha_i \mathbf{h}_i. \tag{3}$$

We can observe that the context vector $\mathbf{c}_t$ is the weighted sum of all the visit information from time 1 to t.

• *Graph-based Attention*. Graph-based attention [7] is proposed to learn robust representations of diagnosis codes even when the data volume is constrained, which explicitly employs the *parent-child* relationship among diagnosis codes with the given medical ontology to learn code embeddings.

Given a medical ontology $\mathcal{G}$ which is a directed acyclic graph (DAG), each leaf node of $\mathcal{G}$ is a diagnosis code c_i and each non-leaf node belongs to the set $\hat{\mathcal{C}}$. Each leaf node has a basic learnable embedding vector $\mathbf{e}_i \in \mathbb{R}^d$ $(1 \leq i \leq |\mathcal{C}|)$, while $\mathbf{e}_{|\mathcal{C}|+1}, \cdots, \mathbf{e}_{|\mathcal{C}|+|\hat{\mathcal{C}}|}$ represent the basic embeddings of the internal nodes $c_{|\mathcal{C}|+1}, \cdots, c_{|\mathcal{C}|+|\hat{\mathcal{C}}|}$. Let $\mathcal{A}(i)$ be the node set of c_i and its ancestors, then the final embedding of diagnosis code c_i denoted by $\mathbf{g}_i \in \mathbb{R}^d$ can be obtained as follows:

$$\mathbf{g}_i = \sum_{j \in \mathcal{A}(i)} \alpha_{ij} \mathbf{e}_j, \quad \sum_{j \in \mathcal{A}(i)} \alpha_{ij} = 1, \tag{4}$$

where

$$\alpha_{ij} = \frac{\exp(\theta(\mathbf{e}_i, \mathbf{e}_j))}{\sum_{k \in \mathcal{A}(i)} \exp(\theta(\mathbf{e}_i, \mathbf{e}_k))}.$$

$\theta(\cdot, \cdot)$ is a scalar value and defined as

$$\theta(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{u}_a^\top \tanh(\mathbf{W}_a \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_j \end{bmatrix} + \mathbf{b}_a),$$

where $\mathbf{u}_a \in \mathbb{R}^l$, $\mathbf{W}_a \in \mathbb{R}^{l \times 2d}$ and $\mathbf{b}_a \in \mathbb{R}^l$ are parameters to be learned. Finally, graph-based attention mechanism generates the medical code embeddings $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{|\mathcal{C}|}\} \in \mathbb{R}^{d \times |\mathcal{C}|}$.

[1] https://en.wikipedia.org/wiki/List_of_ICD-9_codes
[2] http://www.icd9data.com/

Base Models

Since the proposed framework is general, all the existing diagnosis prediction approaches can be cast into this framework and treated as base models. Table I shows the summary of all the state-of-the-art approaches with the aforementioned techniques. The detailed implementation of these base models is introduced in the following section.

TABLE I
BASE MODELS FOR DIAGNOSIS PREDICTION.

Base Model	Visit Modeling		Attention Mechanism	
	FC	GRU	Location	Graph
MLP	√			
RNN [2], [7], [8]		√		
RNN$_a$ [2]		√	√	
Dipole [2]		√	√	
RETAIN [8]		√	√	
GRAM [7]		√		√

C. The Proposed Framework

Different from graph-based attention mechanism which specifies the relationships of diagnosis codes with the given medical ontology, we aim to learn the diagnosis code embeddings directly from their medical descriptions. The main components of the proposed diagnosis prediction framework are *diagnosis code embedding* and *predictive model*. Diagnosis code embedding component is to learn the medical embeddings with code descriptions, which can embed the visit information into a vector representation. Predictive model component aims to predict the future visit information according to the embedded visit representations. Obviously, the proposed framework can be trained end-to-end. Next, we provide the details of these two components.

Diagnosis Code Embedding

To embed the description of each diagnosis code into a vector representation, Convolutional Neural Networks (CNN) [16] can be employed. The benefit of applying CNN is to utilize layers with convolving filters to extract local features, which has shown its superior ability for natural language processing tasks, such as sentence modeling [17] and sentence classification [18].

Figure 1 shows the variant of the CNN architecture to embed each diagnosis code description c_i' into a vector representation $\mathbf{e}_i$. We first obtain the pre-trained embedding of each word w_j denoted as $\mathbf{l}_j \in \mathbb{R}^k$ from fastText [15], where k is the dimensionality. The description c_i' with length n (padded where necessary) is represented as

$$\mathbf{l}_{1:n} = \mathbf{l}_1 \oplus \mathbf{l}_2 \oplus \cdots \oplus \mathbf{l}_n, \tag{5}$$

where $\oplus$ is the concatenation operator. Let h denote the size of a word window, and then $\mathbf{l}_{i:i+h-1}$ represents the concatenation of h words from $\mathbf{l}_i$ to $\mathbf{l}_{i+h-1}$. A filter $\mathbf{W}_f \in \mathbb{R}^{h\times k}$ is applied on the window of h words to produce a new feature $f_i \in \mathbb{R}$ with the ReLU activation function as follows:

$$f_i = \text{ReLU}(\mathbf{W}_f \mathbf{l}_{i:i+h-1} + b_f), \tag{6}$$

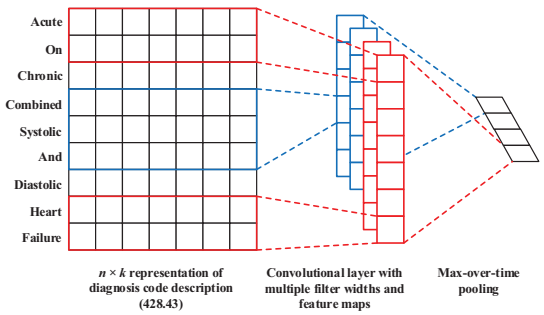

Fig. 1. An Example of CNN Architecture for Diagnosis Code Embedding. The word window sizes are 2 (red line) and 3 (blue line) respectively, i.e., $q = 2$. For each word window, there are 2 filters in the example, i.e., $m = 2$. The dimensionality of this code embedding is 4, i.e., $d = mq = 4$.

where $b_f \in \mathbb{R}$ is a bias term, and $\text{ReLU}(f) = \max(f, 0)$. This filter is applied to each possible window of words in the whole description $\{\mathbf{l}_{1:h}, \mathbf{l}_{2:h+1}, \cdots, \mathbf{l}_{n-h+1:n}\}$ to generate a feature map $\mathbf{f} \in \mathbb{R}^{n-h+1}$ as follows:

$$\mathbf{f} = [f_1, f_2, \cdots, f_{n-h+1}]. \tag{7}$$

Next, max pooling technique [19] is used over the feature map to obtain the most important feature, i.e., $\hat{f} = \max(\mathbf{f})$. In this way, one filter produces one feature. To obtain multiple features, we use m filters with varying window sizes. Here, we use q to denote the number of different window sizes. All the extracted features are concatenated to represent the embedding of each diagnosis code $\mathbf{e}_i \in \mathbb{R}^d$ ($d = mq$). Finally, we can obtain the diagnosis code embedding matrix $\mathbf{E} \in \mathbb{R}^{d\times|\mathcal{C}|}$, where $\mathbf{e}_i$ is the i-th column of $\mathbf{E}$.

The advantage of the proposed CNN-based diagnosis code embedding approach is that it easily makes the diagnosis codes with similar meanings obtain similar vector representations. Thus, for those diagnosis codes without sufficient training EHR data, they still can learn reasonable vector representations, which further helps the model to improve the predictive performance. In the following, we will introduce how to use the produced medical embeddings for the diagnosis prediction task.

Predictive Model

Based on the learned diagnosis code embedding matrix $\mathbf{E}$, we can predict patients' future visit information with a predictive model. Given a visit $\mathbf{x}_t \in \{0, 1\}^{|\mathcal{C}|}$, we first embed $\mathbf{x}_t$ into a vector representation $\mathbf{v}_t \in \mathbb{R}^d$ with $\mathbf{E}$ as follows:

$$\mathbf{v}_t = \tanh(\mathbf{E}\mathbf{x}_t + \mathbf{b}_v), \tag{8}$$

where $\mathbf{b}_v \in \mathbb{R}^d$ is the bias vector to be learned. Then $\mathbf{v}_t$ is fed into the predictive model to predict the $(t+1)$-th visit information, i.e., $\hat{\mathbf{y}}_t$. Next, we cast state-of-the-art diagnosis prediction approaches into the proposed framework as the predictive models.

• *Enhanced MLP* (MLP+). The simplest predictive model is only using a Multilayer Perceptron (MLP) with two layers: a fully-connected layer and a softmax layer, i.e.,

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{h}_t), \tag{9}$$

where $\mathbf{h}_t$ is obtained from Eq. (1). This model works well when both the number of diagnosis codes and patients' visits are small. However, MLP+ does not use historical visit information for the prediction. To overcome the shortage of MLP+, we employ Recurrent Neural Networks (RNN) to handle more complicated scenarios.

• *Enhanced RNN* (RNN+). For RNN+, the visit embedding vector $\mathbf{v}_t$ is fed into a GRU, which produces a hidden state $\mathbf{h}_t \in \mathbb{R}^g$ as follows:

$$\mathbf{h}_t = \text{GRU}(\mathbf{v}_t; \Omega). \tag{10}$$

Then the hidden state $\mathbf{h}_t$ is fed through the softmax layer to predict the $(t+1)$-th visit information as follows:

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_c \mathbf{h}_t + \mathbf{b}_c), \tag{11}$$

where $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}| \times g}$ and $\mathbf{b}_c \in \mathbb{R}^{|\mathcal{C}|}$. Note that RNN+ only uses the t-th hidden state to make the prediction, which does not utilize the information of visits from time 1 to $t-1$. To consider all the information before the prediction, attention-based models are proposed in the following.

• *Enhanced Attention-based RNN* (RNN$_a$+). According to Eq. (10), we can obtain all the hidden states $\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_t$. Then location-based attention mechanism is applied to obtain the context vector $\mathbf{c}_t$ with Eq. (3). Finally, the context vector $\mathbf{c}_t$ is fed into the softmax layer to make predictions as follows:

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_c \mathbf{c}_t + \mathbf{b}_c). \tag{12}$$

• *Enhanced Dipole* (Dipole+). Actually, one drawback of RNN is that prediction performance will drop when the length of sequence is very large [20]. To overcome this drawback, Dipole [2], which uses bidirectional recurrent networks (BRNN) with attention mechanisms, is proposed to improve the prediction performance.

Given the visit embeddings from $\mathbf{v}_1$ to $\mathbf{v}_t$, a BRNN can learn two sets of hidden states: forward hidden states $\overrightarrow{\mathbf{h}}_1, \cdots, \overrightarrow{\mathbf{h}}_t$ and backward hidden states $\overleftarrow{\mathbf{h}}_1, \cdots, \overleftarrow{\mathbf{h}}_t$. By concatenating $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$, we can obtain the final hidden state $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]^\top$ ($\mathbf{h}_t \in \mathbb{R}^{2g}$). Then location-based attention mechanism is used to produce the context vector $\mathbf{c}_t \in \mathbb{R}^{2g}$ with Eq. (2) ($\mathbf{W}_\alpha \in \mathbb{R}^{2g}$). With the learned $\mathbf{c}_t$, Dipole+ can predict the $(t+1)$-th visit information with a softmax layer, i.e., Eq. (12) with $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}| \times 2g}$.

• *Enhanced RETAIN* (RETAIN+). RETAIN [8] is an interpretable diagnosis prediction model, which uses two *reverse time-ordered* GRUs and attention mechanisms to calculate the contribution scores of all the appeared diagnosis codes before the prediction.

The visit-level attention scores can be obtained using Eq. (2). For the code-level attention scores, RETAIN employs the following function:

$$\beta_t = \tanh(\mathbf{W}_\beta \mathbf{h}_t + \mathbf{b}_\beta), \tag{13}$$

where $\mathbf{W}_\beta \in \mathbb{R}^{d \times g}$ and $\mathbf{b}_\beta \in \mathbb{R}^d$ are parameters. Then the context vector $\mathbf{c}_t \in \mathbb{R}^d$ is obtained as follows:

$$\mathbf{c}_t = \sum_{i=1}^{t} \alpha_i \beta_i \circ \mathbf{v}_i. \tag{14}$$

With the generated context vector $\mathbf{c}_t$ and Eq. (12) ($\mathbf{W}_c \in \mathbb{R}^d$), RETAIN+ can predict the $(t+1)$-th patient's health status.

• *Enhanced GRAM* (GRAM+). GRAM [7] is the state-of-the-art approach to learn reasonable and robust representations of diagnosis codes with medical ontologies. To enhance GRAM with the proposed framework, instead of randomly assigning the basic embedding vectors $\mathbf{e}_1, \cdots, \mathbf{e}_{|\mathcal{C}|}$, we use diagnosis code descriptions to learn those embeddings, i.e., $\mathbf{E}$. Note that the non-leaf nodes are still randomly assigned basic embeddings.

With the learned diagnosis code embedding matrix $\mathbf{G}$ as described in Section II-B, we can obtain visit-level embedding $\mathbf{v}_t$ with Eq. (8) (i.e., replacing $\mathbf{E}$ to $\mathbf{G}$). Using Eq. (10) and Eq. (11), GRAM+ predicts the $(t+1)$-th visit information.

III. Experiments

A. Real-World Datasets

Two medical claim datasets are used in our experiments to validate the proposed framework, which are the MIMIC-III dataset and the Heart Failure dataset.

• The MIMIC-III dataset, a publicly available EHR dataset, consists of medical records of 7,499 intensive care unit (ICU) patients over 11 years. For this dataset, we chose the patients who made at least two visits.

• The Heart Failure dataset is an insurance claim dataset, which has 4,925 patients and 341,865 visits from the year 2004 to 2015. The patient visits were grouped by week [2], and we chose patients who made at least two visits. Table II shows more details about the two datasets.

Diagnosis prediction task aims to predict the diagnosis information of the next visit. In our experiments, we intend to predict the diagnosis categories as [2], [7], instead of predicting the real diagnosis codes. Predicting category information not only increases the training speed and predictive performance, but also guarantees the sufficient granularity of all the diagnoses. The nodes in the second hierarchy of the ICD9 codes are used as the category labels [3]. For example, the category label of diagnosis code "428.43: Acute on chronic combined systolic and diastolic heart failure" is "Diseases of the circulatory system (390-459)".

B. Experimental Setup

We first introduce the state-of-the-art diagnosis prediction approaches as base models, then describe the measures to evaluate the prediction results of all the approaches, and finally present the details of our experiment implementation.

[3]Note that the hierarchy of CCS (https://www.hcup-us.ahrq.gov/toolssoftware/ccs/AppendixASingleDX.txt) can also be used as category labels [7]. These two kinds of grouping methods can obtain similar predictive performance.

TABLE II
STATISTICS OF MIMIC-III AND HEART FAILURE DATASETS.

Dataset	MIMIC-III	Heart Failure
# of patients	7,499	4,925
# of visits	19,911	341,865
Avg. visits per patient	2.66	69.41
# of unique ICD9 codes	4,880	6,747
Avg. # of diagnosis codes per visit	13.06	3.92
Max # of diagnosis codes per visit	39	54
# of words in code descriptions	2,800	3,397
# of category codes	171	149
Avg. # of category codes per visit	10.16	3.33
Max # of category codes per visit	30	33

Base Models

In our experiments, we use the following six approaches as base models: MLP, RNN, RNN_a [2], Dipole [2], RETAIN [8], and GRAM [7]. For all the base models, we all design the corresponding enhanced approaches for comparison.

Evaluation Measures

To fairly evaluate the performance of all the diagnosis prediction approaches, we validate the results with the measure *accuracy*@k. Given a visit V_t which contains multiple category labels, if the target label is in the top k guesses, then we get 1 and 0 otherwise. Thus, *accuracy*@k is defined by the number of correct label predictions divided by the total number of label predictions. The greater values, the better performance. In the experiments, we vary k from 5 to 30.

Implementation Details

We extract the diagnosis code descriptions from ICD9Data.com. All the approaches are implemented with Theano 0.9.0 [21]. We randomly divide the datasets into the training, validation and testing sets in a 0.75:0.10:0.15 ratio. The validation set is used to determine the best values of parameters in the 100 training iterations. For training models, we use Adadelta [22] with a min-batch of 100 patients. The regularization (l_2 norm with the coefficient 0.001) is used for all the approaches. In order to fairly compare the performance, we set the same $g = 128$ (i.e., the dimensionality of hidden states) for all the base models and the enhanced approaches except MLP and MLP+. For the proposed approaches on both datasets, the size of word embeddings is 300, the word windows (h's) are set as 2, 3 and 4, and thus $q = 3$. For each word window, we use $m = 100$ filters. For all the base models, we set $d = 180$ on the MIMIC-III dataset and 150 on the Heart Failure dataset. For GRAM, l is 100.

C. Results of Diagnosis Prediction

Table III lists the *accuracy* with different k's. We can observe that the enhanced diagnosis prediction approaches improve the prediction performance on both the MIMIC-III and Heart Failure datasets.

Performance Analysis for the MIMIC-III Dataset

On the MIMIC-III dataset, the overall performance of all the enhanced diagnosis prediction approaches is better than that of all the base models. Among all the proposed approaches, RETAIN+ and MLP+ achieve higher accuracy. MLP+ does not use recurrent neural networks and directly predicts the future diagnosis information with the learned visit embedding $\mathbf{v}_t$. RETAIN+ utilizes the context vector which learns from visit-level and code-level attention scores, and the learned visit embeddings to make the final predictions. However, all the remaining proposed approaches use the hidden states outputted from GRUs to predict the next visit information. From the above analysis, we can conclude that directly adding visit embeddings into the final prediction can improve the predictive performance on the MIMIC-III dataset. This is reasonable because the average length of visits is small on the MIMIC-III dataset. The shorter visits may not help the RNN-based models to learn correct hidden states, and thus those methods can not achieve the highest accuracy.

This observation can also be found from the performance of all the base models. Compared with the naive base model MLP, the precision or accuracy of all the four RNN-based approaches is lower, including RNN, RNN_a, Dipole and RETAIN. This again confirms that RNN-based models cannot work well with short sequences. Among all the RNN-based approaches, location-based attention models, RNN_a and Dipole, perform worse than RNN and RETAIN, which shows that learning attention mechanisms needs abundant EHR data. Compared with RNN, both the precision and accuracy of RETAIN are still higher. This demonstrates that directly using visit embedding in the final prediction may achieve better performance for the datasets with shorter visit sequences. GRAM can achieve comparable performance with the naive base model MLP. It proves that employing external information can compensate for the lack of training EHR data in diagnosis prediction task.

Performance Analysis for the Heart Failure Dataset

On the Heart Failure dataset, the enhanced approaches still perform better than the corresponding base models, especially GRAM+ which achieves much higher accuracy than other approaches. The reason is that GRAM+ not only uses medical ontologies to learn robust diagnosis code embeddings, but also employs code descriptions to further improve the performance, which can be validated from the comparison between the performance of GRAM and GRAM+.

Among all the approaches, the accuracy of RETAIN is the lowest, which shows that directly using the visit-level embeddings in the final prediction may not work on the Heart Failure dataset, which can also be observed from the performance of MLP. However, taking code descriptions into consideration, the performance enormously increases. When $k = 5$, the accuracy of RETAIN improves 42%. The performance of MLP is better than that of RETAIN, but it is still lower than other RNN variants. This illustrates that with complicated EHR datasets, simply using multilayer perceptrons cannot work well. Though learning medical embeddings of diagnosis codes improves the predictive performance, the accuracy of MLP+ is still lower than that of most approaches. This directly validates that applying recurrent neural networks to diagnosis prediction task is reasonable.

TABLE III
RESULTS OF DIAGNOSIS PREDICTION TASK.

Dataset	@k	MLP	MLP+	RNN	RNN+	RNN$_a$	RNN$_a$+	Dipole	Dipole+	RETAIN	RETAIN+	GRAM	GRAM+
MIMIC-III	5	0.3104	**0.3181**	0.2952	**0.3193**	0.2910	**0.3162**	0.2941	**0.3155**	0.3056	**0.3198***	0.3072	**0.3183**
	10	0.5040	**0.5138**	0.4796	**0.5111**	0.4693	**0.5085**	0.4767	**0.5086**	0.4980	**0.5160***	0.5003	**0.5138**
	15	0.6286	**0.6352**	0.6019	**0.6335**	0.5889	**0.6290**	0.5971	**0.6325**	0.6258	**0.6360***	0.6267	**0.6348**
	20	0.7114	**0.7239***	0.6894	**0.7198**	0.6822	**0.7144**	0.6845	**0.7168**	0.7129	**0.7202**	0.7130	**0.7196**
	25	0.7754	**0.7852***	0.7545	**0.7804**	0.7491	**0.7785**	0.7501	**0.7795**	0.7735	**0.7806**	0.7728	**0.7794**
	30	0.8214	**0.8294***	0.8040	**0.8279**	0.7987	**0.8269**	0.7990	**0.8280**	0.8198	**0.8286**	0.8220	**0.8283**
Heart Failure	5	0.4580	**0.5132**	0.5599	**0.5960**	0.5699	**0.5882**	0.5687	**0.5868**	0.4085	**0.5808**	0.6152	**0.6227***
	10	0.6266	**0.6412**	0.6835	**0.7169**	0.6920	**0.7109**	0.6953	**0.7105**	0.5460	**0.7042**	0.7393	**0.7455***
	15	0.7124	**0.7254**	0.7603	**0.7876**	0.7645	**0.7845**	0.7702	**0.7841**	0.6512	**0.7765**	0.8088	**0.8130***
	20	0.7717	**0.7827**	0.8132	**0.8355**	0.8153	**0.8334**	0.8209	**0.8307**	0.7162	**0.8261**	0.8544	**0.8580***
	25	0.8206	**0.8283**	0.8516	**0.8698**	0.8532	**0.8673**	0.8580	**0.8655**	0.7684	**0.8622**	0.8872	**0.8902***
	30	0.8572	**0.8635**	0.8812	**0.8958**	0.8825	**0.8943**	0.8860	**0.8923**	0.8100	**0.8899**	0.9113	**0.9134***

* denotes the highest accuracy among all the approaches on the same k.

For the two location-based attention approaches, RNN$_a$ and Dipole, the performance is better than that of RNN, which demonstrates that attention mechanisms can help the models to enhance the predictive ability. Comparison between RNN$_a$ and Dipole confirms that when the length of visit sequences is large, bidirectional recurrent neural networks can remember more useful information and perform better than one directional recurrent neural networks.

Based on all the above analysis, we can safely conclude that learning diagnosis code embeddings with descriptions indeed helps all the state-of-the-art diagnosis prediction approaches to significantly improve the performance on different real world datasets.

IV. CONCLUSIONS

In this paper, we propose a novel and effective diagnosis prediction framework, which takes the medical meanings of diagnosis codes into account when predicting patients' future visit information. Experimental results on two real world medical datasets prove the effectiveness and robustness of the proposed framework for diagnosis prediction task.

ACKNOWLEDGMENT

This work is supported in part by the US National Science Foundation under grant IIS-1747614. The authors would like to thank NVIDIA Corporation with the donation of the Titan Xp GPU. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," *Briefings in Bioinformatics*, p. bbx044, 2017.

[2] 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 *KDD*, 2017, pp. 1903–1911.

[3] F. Ma, G. Jing, Q. Suo, Q. You, J. Zhou, and A. Zhang, "Risk prediction on electronic health records with prior medical knowledge," in *KDD*. ACM, 2018, pp. 1910–1919.

[4] F. Ma, C. Meng, H. Xiao, Q. Li, J. Gao, L. Su, and A. Zhang, "Unsupervised discovery of drug side-effects from heterogeneous data sources," in *KDD*, 2017, pp. 967–976.

[5] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, and J. Gao, "Kame: Knowledge-based attention model for diagnosis prediction in healthcare," in *CIKM*. ACM, 2018, pp. 743–752.

[6] 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 *KDD*, 2016, pp. 1495–1504.

[7] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: Graph-based attention model for healthcare representation learning," in *KDD*, 2017, pp. 787–795.

[8] 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 *NIPS*, 2016, pp. 3504–3512.

[9] Y. Yuan, G. Xun, F. Ma, Q. Suo, H. Xue, K. Jia, and A. Zhang, "A novel channel-aware attention framework for multi-channel eeg seizure detection via multi-view deep learning," in *BHI*. IEEE, 2018, pp. 206–209.

[10] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, J. Gao, and A. Zhang, "Deep patient similarity learning for personalized healthcare," *IEEE Transactions on NanoBioscience*, 2018.

[11] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, and A. Zhang, "Personalized disease prediction using a cnn-based similarity learning method," in *BIBM*, 2017.

[12] Q. Suo, F. Ma, G. Canino, J. Gao, A. Zhang, P. Veltri, and A. Gnasso, "A multi-task framework for monitoring health conditions via attention-based recurrent neural networks," in *AMIA*, 2017.

[13] Y. Yuan, G. Xun, F. Ma, Y. Wang, N. Du, K. Jia, L. Su, and A. Zhang, "Muvan: A multi-view attention network for multivariate temporal data," in *ICDM*. IEEE, 2018.

[14] Q. Suo, W. Zhong, F. Ma, Y. Yuan, M. Huai, and A. Zhang, "Multi-task sparse metric learning for monitoring patient similarity progression," in *ICDM*, 2018.

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] P. Blunsom, E. Grefenstette, and N. Kalchbrenner, "A convolutional neural network for modelling sentences," in *ACL*, 2014, pp. 655–665.

[18] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP*, 2014, pp. 1746–1751.

[19] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] T. T. D. Team, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[22] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.