Ma *et al. BMC Medical Informatics and Decision Making* 2019, **19**(Suppl 6):267
https://doi.org/10.1186/s12911-019-0961-2

BMC Medical Informatics and Decision Making

RESEARCH Open Access

Incorporating medical code descriptions for diagnosis prediction in healthcare

Fenglong Ma[1*], Yaqing Wang[2], Houping Xiao[3], Ye Yuan[4], Radha Chitta[5], Jing Zhou[6] and Jing Gao[2]

From IEEE International Conference on Bioinformatics and Biomedicine 2018
Madrid, Spain. 3-6 December 2018

Abstract

Background: Diagnosis prediction aims to predict the future health status of patients according to their historical electronic health records (EHR), which is an important yet challenging task in healthcare informatics. Existing diagnosis prediction approaches mainly employ recurrent neural networks (RNN) 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 meaning code representations, but also to improve the predictive performance, especially when the EHR data are insufficient.

Methods: 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 (CNN) 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).

Results: 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. Moreover, we validate the effectiveness of the proposed framework with insufficient EHR data. Finally, we visualize the learned medical code embeddings to show the interpretability of the proposed framework.

Conclusions: Given the historical visit records of a patient, the proposed framework is able to predict the next visit information by incorporating medical code descriptions.

Keywords: Healthcare informatics, Diagnosis prediction, Medical code embeddings

Background

The immense accumulation of Electronic Healthcare Records (EHR) makes it possible to directly predict patients' future health status, which is done by analyzing their historical visit records [1–4]. *Diagnosis prediction* attracts considerable attention from both healthcare providers and researchers. It aims to predict the diagnosis information of patients in the following visits. There are two key challenges for diagnosis prediction task as follows: (1) designing an accurate and robust predictive model to handle the temporal, high dimensional and noisy EHR data; and (2) reasonably interpreting the advantages and effectiveness of the proposed models to both doctors and patients.

To address these challenges of diagnosis prediction task, many recurrent neural networks (RNN) based models [2–4] have been proposed. RETAIN [4] uses two recurrent neural networks with attention mechanisms to model the reverse time ordered EHR sequences. By employing a bidirectional recurrent neural network (BRNN), Dipole [2] enhances the prediction accuracy with different attention mechanisms. In order to guarantee the predictive performance, training the above mentioned models usually

*Correspondence: fenglong@psu.edu
[1]Pennsylvania State University, State College, PA, USA
Full list of author information is available at the end of the article

requires a lot of EHR data. However, there is a common problem for EHR data that is always existing medical codes of rare diseases. Those diagnosis codes infrequently appear in the EHR data. GRAM [3] has been proposed to overcome this issue. GRAM 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, the description of diagnosis code "428.32" is "*Chronic diastolic heart failure*" (http://www.icd9data.com/2015/Volume1/390-459/420-429/428/428.32.htm), and "*Rheumatic heart failure (congestive)*" is the description of diagnosis code "398.91" (http://www.icd9data.com/2015/Volume1/390-459/393-398/398/398.91.htm). 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** in the prediction should help the predictive models to improve the prediction accuracy and provide interpretable representations of medical codes, especially when the EHR data are insufficient.

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 [5]. Then the concatenation of all the words 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.

We use two real medical datasets to illustrate the superior ability of the proposed framework on the diagnosis prediction task compared with several state-of-the-art approaches. Quantitative analysis is also conducted to validate the effectiveness of the proposed approaches with insufficient EHR data. Finally, we qualitatively analyze the interpretability of the enhanced approaches by visualizing the learned medical code embeddings against the embeddings learned by existing approaches. To sum up, we achieve the following contributions in this paper:

- We realize the importance of obtaining diagnosis code embeddings from their descriptions which can be directly extracted from the Internet.
- We propose a simple, but general and effective diagnosis prediction framework, which learns representations of diagnosis codes directly from their descriptions.
- All the state-of-the-art approaches can be cast into the proposed framework to improve the performance of diagnosis prediction.
- Experimental results on two medical datasets validate the effectiveness of the proposed framework and the interpretability for prediction results.

Related Work

In this section, we briefly survey the work related to diagnosis prediction task. We first provide a general introduction about mining healthcare related data with deep learning techniques, and then survey the work of diagnosis prediction.

Deep Learning for EHR

Several machine learning approaches are proposed to mine medical knowledge from EHR data [1, 6–10]. Among them, deep learning-based models have achieved better performance compared with traditional machine learning approaches [11–13]. To detect the characteristic patterns of physiology in clinical time series data, stacked denoising autoencoders (SDA) are used in [14]. Convolutional neural networks (CNN) are applied to predict unplanned readmission [15], sleep stages [16], diseases [17, 18] and risk [19–21] with EHR data. To capture

the temporal characteristics of healthcare related data, recurrent neural networks (RNN) are widely used for modeling disease progression [22, 23], mining time series healthcare data with missing values [24, 25], and diagnosis classification [26] and prediction [2–4, 27].

Diagnosis Prediction

Diagnosis prediction is one of the core research tasks in EHR data mining, which aims to predict the future visit information according to the historical visit records. Med2Vec [28] is the first unsupervised method to learn the interpretable embeddings of medical codes, but it ignores long-term dependencies of medical codes among visits. RETAIN [4] is the first interpretable model to mathematically calculate the contribution of each medical code to the current prediction by employing a reverse time attention mechanism in an RNN for binary prediction task. Dipole [2] is the first work to adopt bidirectional recurrent neural networks (BRNN) and different attention mechanisms to improve the prediction accuracy. GRAM [3] is the first work to apply graph-based attention mechanism on the given medical ontology to learn robust medical code embeddings even when lack of training data, and an RNN is used to model patient visits. KAME [29] uses high-level knowledge to improve the predictive performance, which is build upon GRAM.

However, different from all the aforementioned diagnosis prediction models, the proposed diagnosis prediction framework incorporates the descriptions of diagnosis codes to learn embeddings, which greatly improves the prediction accuracy and provide interpretable prediction results against the state-of-the-art approaches.

Methods

In this section, we first mathematically define the notations used in the diagnosis prediction task, introduce preliminary concepts, and then describe the details of the proposed framework.

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 $\mathcal{V}^{(p)} = \left\{V_1^{(p)}, V_2^{(p)}, \cdots, V_T^{(p)}\right\}$. 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 (https://en.wikipedia.org/wiki/List_of_ICD-9_codes) or ICD9Data.com (http://www.icd9data.com/). 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 $\left\{c_1', c_2', \cdots, c_{|\mathcal{C}|}'\right\}$ and the set of time-ordered sequences of patient visits $\left\{\mathbf{x}_1^{(p)}, \mathbf{x}_2^{(p)}, \cdots, \mathbf{x}_{T-1}^{(p)}\right\}_{p=1}^{|\mathcal{P}|}$. For each timestep t, we aim to predict the information of the $(t+1)$-th visit. Thus, the outputs are $\left\{\mathbf{x}_2^{(p)}, \mathbf{x}_3^{(p)}, \cdots, \mathbf{x}_T^{(p)}\right\}_{p=1}^{|\mathcal{P}|}$.

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 [2–4, 30]. Note that we use "RNN" to denote any Recurrent Neural Network variants, such as Long-Short Term Memory (LSTM) [31], T-LSTM [32] and Gated Recurrent Unit (GRU) [33]. In this paper, GRU is used to adaptively capture dependencies among patient visit information. GRU has two gates: One is the reset gate r, and the other is the update gate z. The reset gate r computes its state from both the new input and the previous memory. The function of r is to make the hidden layer drop irrelevant information. The update gate z controls how much information should be kept around from the previous hidden state. The mathematical formulation of GRU can be described as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{v}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \beta_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h \beta_t + \mathbf{r}_t \circ \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h), \\
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (\mathbf{1} - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t,
\end{aligned} \tag{2}$$

where $\mathbf{z}_t \in \mathbb{R}^g$ is the update gate at time t, g is the dimensionality of hidden states, $\sigma()$ is the activation function,

$\mathbf{h}_t \in \mathbb{R}^g$ is the hidden state, $\mathbf{r}_t \in \mathbb{R}^g$ is the reset gate at time t, $\tilde{\mathbf{h}}_t \in \mathbb{R}^g$ represents the intermediate memory, and $\circ$ denotes the element-wise multiplication. Matrices $\mathbf{W}_z \in \mathbb{R}^{g\times d}$, $\mathbf{W}_r \in \mathbb{R}^{g\times d}$, $\mathbf{W}_h \in \mathbb{R}^{g\times d}$, $\mathbf{U}_z \in \mathbb{R}^{g\times g}$, $\mathbf{U}_r \in \mathbb{R}^{g\times g}$, $\mathbf{U}_h \in \mathbb{R}^{g\times g}$ and vectors $\mathbf{b}_z \in \mathbb{R}^g$, $\mathbf{b}_r \in \mathbb{R}^g$, $\mathbf{b}_h \in \mathbb{R}^g$ are parameters to be learned. For simplicity, the GRU can be represented by

$$\mathbf{h}_t = \text{GRU}(\beta_t; \Omega), \tag{3}$$

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, 4], general attention [2], concatenation-based attention [2], and graph-based attention [3]. In the following, we introduce two commonly used attention mechanisms: location-based and graph-based attention.

• *Location-based Attention.* Location-based attention mechanism [2, 4] is to calculate the attention score for each visit, which solely depends on the current hidden state $\mathbf{h}_i \in \mathbb{R}^g$ $(1 \le i \le t)$ as follows:

$$\alpha_i = \mathbf{W}_\alpha^\top \mathbf{h}_i + b_\alpha, \tag{4}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^g$ and $b_\alpha \in \mathbb{R}$ are the parameters to be learned. According to Eq. (4), 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:

$$\mathbf{c}_t = \sum_{i=1}^{t} \alpha_i \mathbf{h}_i. \tag{5}$$

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 [3] 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 $\mathbf{e}_i \in \mathbb{R}^d$ $(1 \le i \le |\mathcal{C}|)$, 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:

$$\mathbf{g}_i = \sum_{j\in\mathcal{A}(i)} \alpha_{ij}\mathbf{e}_j, \quad \sum_{j\in\mathcal{A}(i)} \alpha_{ij} = 1, \tag{6}$$

where

$$\alpha_{ij} = \frac{\exp(\theta(\mathbf{e}_i, \mathbf{e}_j))}{\sum_{k\in\mathcal{A}(i)} \exp(\theta(\mathbf{e}_i, \mathbf{e}_k))}. \tag{7}$$

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

$$\theta(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{u}_a^\top \tanh\left(\mathbf{W}_a \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_j \end{bmatrix} + \mathbf{b}_a\right), \tag{8}$$

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 $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{|\mathcal{C}|}\} \in \mathbb{R}^{d\times|\mathcal{C}|}$.

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 1 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 "Experimental Setup" section.

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.

Table 1 Base models for diagnosis prediction

Base model	Visit modeling		Attention mechanism	
	FC	GRU	Location	Graph
MLP	√			
RNN [2–4]		√		
RNN$_\alpha$ [2]		√	√	
Dipole [2]		√	√	
RETAIN [4]		√	√	
GRAM [3]		√		√

Diagnosis Code Embedding

To embed the description of each diagnosis code into a vector representation, Convolutional Neural Networks (CNN) [34] 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 [35] and sentence classification [36].

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 [5], where k is the dimensionality. The description c'_i with length n (padded where necessary) is represented as

$$\mathbf{l}_{1:n} = \mathbf{l}_1 \oplus \mathbf{l}_2 \oplus \cdots \oplus \mathbf{l}_n, \tag{9}$$

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:

$$f_i = \text{ReLU}(\mathbf{W}_f \mathbf{l}_{i:i+h-1} + b_f), \tag{10}$$

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:

$$\mathbf{f} = [f_1, f_2, \cdots, f_{n-h+1}]. \tag{11}$$

Next, max pooling technique [37] 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 $\mathbf{E} \in \mathbb{R}^{d\times|C|}$, 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 $\mathbf{x}_t \in \{0, 1\}^{|C|}$, we first embed $\mathbf{x}_t$ into a vector representation $\mathbf{v}_t \in \mathbb{R}^d$ with $\mathbf{E}$ as follows:

$$\mathbf{v}_t = \tanh(\mathbf{E}\mathbf{x}_t + \mathbf{b}_v), \tag{12}$$

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.,

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{h}_t), \tag{13}$$

where $\mathbf{h}_t$ is obtained from Eq. (1). This model works well when both the number of diagnosis codes and patients'

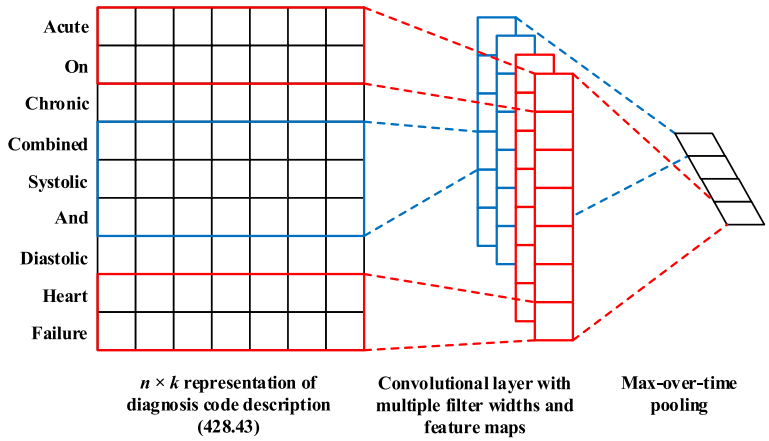

Fig. 1 An Example of CNN Architecture for Diagnosis Code Embedding. The word window sizes are 2 (red line) and 3 (blue line) respectively, i.e., $q = 2$. For each word window, there are 2 filters in the example, i.e., $m = 2$. The dimensionality of this code embedding is 4, i.e., $d = mq = 4$

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:

$$\mathbf{h}_t = \text{GRU}(\mathbf{v}_t; \Omega). \tag{14}$$

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

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_c \mathbf{h}_t + \mathbf{b}_c), \tag{15}$$

where $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}| \times g}$. 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. (14), 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. (5). Finally, the context vector $\mathbf{c}_t$ is fed into the softmax layer to make predictions as follows:

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_c \mathbf{c}_t + \mathbf{b}_c). \tag{16}$$

• *Enhanced Dipole* (Dipole+). Actually, one drawback of RNN is that prediction performance will drop when the length of sequence is very large [38]. To overcome this drawback, Dipole [2] which uses bidirectional recurrent networks (BRNN) with attention mechanisms are 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. (4) ($\mathbf{W}_\alpha \in \mathbb{R}^{2g}$). With the learned $\mathbf{c}_t$, Dipole+ can predict the $(t+1)$-th visit information with a softmax layer, i.e., Eq. (16) with $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}| \times 2g}$.

• *Enhanced RETAIN* (RETAIN+). RETAIN [4] 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. (4). For the code-level attention scores, RETAIN employs the following function:

$$\beta_t = \tanh(\mathbf{W}_\beta \mathbf{h}_t + \mathbf{b}_\beta), \tag{17}$$

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:

$$\mathbf{c}_t = \sum_{i=1}^{t} \alpha_i \beta_i \circ \mathbf{v}_i. \tag{18}$$

With the generated context vector $\mathbf{c}_t$ and Eq. (16) ($\mathbf{W}_c \in \mathbb{R}^d$), RETAIN+ can predict the $(t+1)$-th patient's health status.

• *Enhanced GRAM* (GRAM+). GRAM [3] 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 "Preliminaries" section, we can obtain visit-level embedding $\mathbf{v}_t$ with Eq. (12) (i.e., replacing $\mathbf{E}$ to $\mathbf{G}$). Using Eqs. (14) and (15), GRAM+ predicts the $(t+1)$-th visit information.

Remark: A key benefit of the proposed framework is its flexibility and transparency relative to all the existing diagnosis prediction models. Beyond all the aforementioned base approaches, more effective and complicated diagnosis prediction models can also be easily cast into the proposed framework.

Results

In this section, we first introduce two real world medical datasets used in the experiments, and then describe the settings of experiments. Finally, we validate the proposed framework on the two datasets.

Real-World Datasets

Two medical claim datasets are used in our experiments to validate the proposed framework, which are the MIMIC-III dataset [39] 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 2 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, 3], instead of predicting the real diagnosis codes. Predicting category information not only increases the training speed and predictive performance, but also guarantees the sufficient

Table 2 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

granularity of all the diagnoses. The nodes in the second hierarchy of the ICD9 codes are used as the category labels. 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)".

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.

Base Models

In our experiments, we use the following six approaches as base models:

- MLP. MLP is a naive method, which first embeds the input visit $\mathbf{x}_t$ into a vector space $\mathbf{v}_t$, and then uses Eq. (1) and Eq. (13) to predict the $(t+1)$-th visit information.
- RNN. RNN is a commonly used model. The input visit is first embedded into a visit-level representation $\mathbf{v}_t$ with a randomly initialized embedding matrix. Then $\mathbf{v}_t$ is fed into a GRU, and the GRU outputs the hidden state $\mathbf{h}_t$ (Eq. (14)), which is used to predict the next visit information with Eq. (15).
- RNN$_a$ [2]. RNN$_a$ adds the location-based attention mechanism into RNN. After the GRU outputs the hidden states $\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_t$, RNN$_a$ employs Eqs. (4) and (5) to calculate the context vector $\mathbf{c}_t$. Finally, RNN$_a$ makes the predictions using the learned $\mathbf{c}_t$ and Eq. (16).
- Dipole [2]. Dipole is the first work to apply bidirectional recurrent neural networks to diagnosis prediction task. In our experiments, we use location-based attention mechanism. Compared with RNN$_a$, the difference is that Dipole uses two GRUs to generate the hidden states, and then concatenates these two sets of hidden states to calculate the context vector $\mathbf{c}_t$ with location-based attention mechanism.
- RETAIN [4]. RETAIN focuses on interpreting the prediction results with a two-level attention model. RETAIN uses a reverse time-ordered visit sequence to calculate the visit-level attention score with Eq. (4). The other GRU is used to compute the code-level attention weight with Eq. (17). The context vector $\mathbf{c}_t$ is obtained using Eq. (18). Based on this context vector, RETAIN predicts the $(t+1)$-th diagnosis codes.
- GRAM [3]. GRAM is the first work to employ medical ontologies to learn diagnosis code representations and predict the future visit information with recurrent neural networks. GRAM first learns the diagnosis code embedding matrix $\mathbf{G}$ with graph-based attention mechanism (Eq. (6)). With the learned $\mathbf{G}$, the input visit $\mathbf{x}_t$ is embedded into a visit-level representation $\mathbf{v}_t$, which is fed into a GRU to produce the hidden state $\mathbf{h}_t$. Equation (15) is used to make the final predictions.

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 from aspects: visit level and code level with the measures *precision@k* and *accuracy@k*.

- *Visit-level precision@k* is defined as the correct diagnosis codes in top k divided by $\min(k, |\mathbf{y}_t|)$, where $|\mathbf{y}_t|$ is the number of category labels in the $(t+1)$-th visit.
- 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, *code-level accuracy@k* is defined by the number of correct label predictions divided by the total number of label predictions.

Visit-level precision@k is used to evaluate the coarse-grained performance, while *code-level accuracy@k* evaluates the fine-grained performance. For all the measures, 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 [40]. 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 [41] 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.

Results of Diagnosis Prediction

Table 3 shows the *visit-level precision* of all the base models and their corresponding enhanced approaches, and Table 4 lists the *code-level accuracy* with different k's. From these two tables, 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.

Here is an interesting observation: As expected, the performance improves as k increases, except the visit-level accuracy on the MIMIC-III dataset, due to the insufficiency of training data. Compared with the labels with abundant data, they obtain lower probabilities in the predictions. Thus, for the visits containing these labels without sufficient data, the number of correct predictions when k is 10 or 15 may be the same with that when $k = 5$. However, they are divided by a bigger $\min(k, |\mathbf{y}_t|)$, which leads to the observation that the average performance is worse than that with $k = 5$.

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

Table 3 The visit-level precision@k of diagnosis prediction task

Dataset	@k	MLP	MLP+	RNN	RNN+	RNN_a	RNN_a+	Dipole	Dipole+	RETAIN	RETAIN+	GRAM	GRAM+
MIMIC-III	5	0.6939	0.7124	0.6616	0.7160	0.6504	0.7083	0.6599	0.7074	0.6835	0.7167*	0.6885	0.7132
	10	0.6441	0.6603	0.6145	0.6565	0.6021	0.6527	0.6116	0.6539	0.6361	0.6623*	0.6424	0.6596
	15	0.6812	0.6926*	0.6546	0.6906	0.6412	0.6856	0.6524	0.6903	0.6777	0.6918	0.6828	0.6918
	20	0.7420	0.7544*	0.7199	0.7511	0.7109	0.7455	0.7159	0.7483	0.7403	0.7501	0.7434	0.7513
	25	0.7939	0.8070*	0.7755	0.8019	0.7697	0.8009	0.7723	0.8020	0.7912	0.8010	0.7941	0.8028
	30	0.8357	0.8460	0.8186	0.8456	0.8142	0.8445	0.8169	0.8453	0.8335	0.8445	0.8377	0.8468*
Heart failure	5	0.4451	0.4947	0.4890	0.5172	0.4976	0.5103	0.4964	0.5111	0.3751	0.5140	0.5341	0.5365*
	10	0.6122	0.6206	0.6585	0.6879	0.6675	0.6817	0.6689	0.6829	0.5378	0.6828	0.7123	0.7159*
	15	0.6996	0.7060	0.7436	0.7683	0.7496	0.7631	0.7514	0.7648	0.6372	0.7613	0.7901	0.7939*
	20	0.7606	0.7643	0.8006	0.8213	0.8050	0.8174	0.8070	0.8167	0.7088	0.8143	0.8402	0.8442*
	25	0.8100	0.8140	0.8425	0.8593	0.8453	0.8560	0.8476	0.8557	0.7655	0.8533	0.8761	0.8789*
	30	0.8477	0.8511	0.8743	0.8879	0.8770	0.8857	0.8785	0.8846	0.8102	0.8826	0.9025	0.9047*

* denotes the highest precision among all the approaches on the same k

Table 4 The code-level accuracy@*k* 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*

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, both precision and accuracy of RETAIN are 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 visit-level precision and code-level accuracy of RETAIN improve 37% and 42% respectively. 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.

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 size of visit sequences is big, 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.

Discussions

The main contribution of this work is to incorporate code descriptions to improve the prediction performance of state-of-the-art models. The experimental results on two real datasets confirm the effective of the proposed framework. Next, we further discuss the performance changes with the degree of data sufficiency and the representations leaned by the proposed framework.

Data Sufficiency

In healthcare, it is hard to collect enough EHR data for those rare diseases. In order to validate the sensitivity of all the diagnosis prediction approaches to data sufficiency, the following experiments are conducted on the MIMIC-III dataset. We first calculate the frequency of category labels appeared in the training data, then rank these labels according to the frequency, and finally divide them into four groups: 0-25, 25-50, 50-75 and 75-100. The category labels in group 0-25 are the most rare ones in the training data, while the labels in group 75-100 are the most common ones. We finally compute the average accuracy of labels in each group. The code-level accuracy@20 on the MIMIC-III dataset is shown in Fig. 2. X-axis denotes all the base models and their corresponding enhanced approaches, and Y-axis represents the average accuracy of the approaches.

From Fig. 2, we can observe that the accuracy of all the enhanced diagnosis prediction approaches is higher than that of all the base models in the first three groups. Even though MLP and RETAIN achieve higher accuracy compared with RNN, RNN_a and Dipole as shown in Table 4, the accuracy of both approaches is 0 in group 0-25. However,

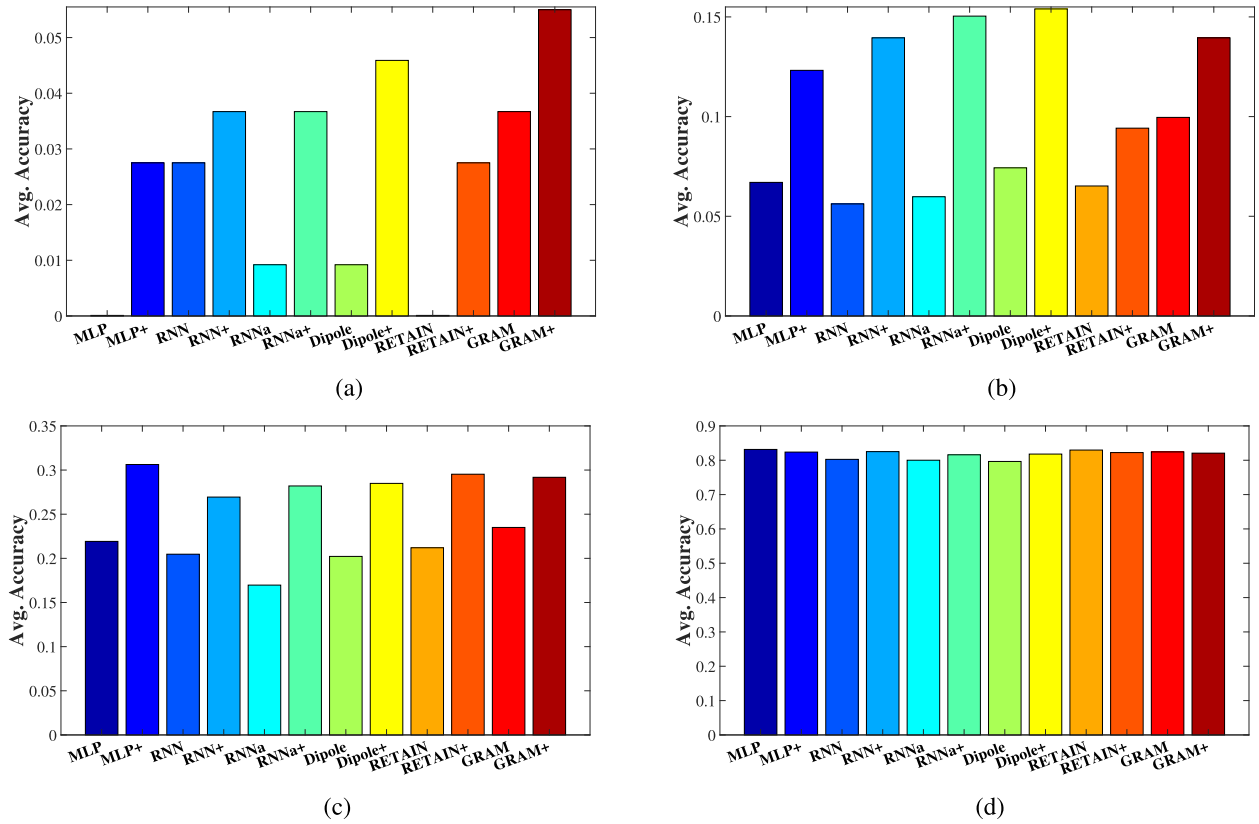

Fig. 2 Code-Level Accuracy@20 of Diagnosis Prediction on the MIMIC-III Dataset. **a** 0-25. **b** 25-50. **c** 50-75. **d** 75-100

when generalizing the proposed framework on MLP and RETAIN, they all make some correct predictions for rare diseases. This observation also can be found in groups 25-50 and 50-70. Therefore, this observation validates that considering the medical meanings of diagnosis codes indeed helps existing models to enhance their predictive ability even without sufficient training EHR data.

In Fig. 2d, all the labels have sufficient and abundant training EHR data. Thus, all the approaches achieve comparable performance. This result again confirms that the enhanced approaches improve the predictive performance on those rare diseases, i.e., the labels without sufficient training EHR records. Among all the base models, GRAM obtains the highest accuracy in groups 0-25, 25-50 and 50-75, which illustrates the effectiveness of incorporating external medical knowledge. Furthermore, learning medical embeddings with ontologies still improves the predictive accuracy, which can be observed from both Fig. 2 and Table 4.

Interpretable Representation

For diagnosis prediction task, interpreting the learned medical code embeddings is significantly important. Thus, we conduct the following experiments to qualitatively demonstrate the learned representations by all the approaches on the MIMIC-III dataset. W randomly select 2000 diagnosis codes and then plot them on a 2-D space with t-SNE [42] shown in Fig. 3. The color of the dots represents the first disease categories in CCS multi-level hierarchy as [3]. We can observe that except GRAM, the remaining baselines cannot learn interpretable representations. However, after considering the semantic meanings learned from diagnosis code descriptions, all the proposed approaches can learn some interpretable cluster structures in the representations. Especially for GRAM+, it not only maintains the advantages of GRAM, but also improves the prediction accuracy. From Fig. 3, we come to a conclusion that the proposed semantic diagnosis prediction framework is effective and interpretable even when the training EHR data are insufficient.

Conclusions

Diagnosis prediction from EHR data is a challenging yet practical research task in healthcare domain. Most state-of-the-art diagnosis prediction models employ recurrent neural networks to model the sequential patients' visit records, and exploit attention mechanisms to improve the predictive performance and provide interpretability for the prediction results. However, all the existing models ignore the medical descriptions of diagnosis codes, which

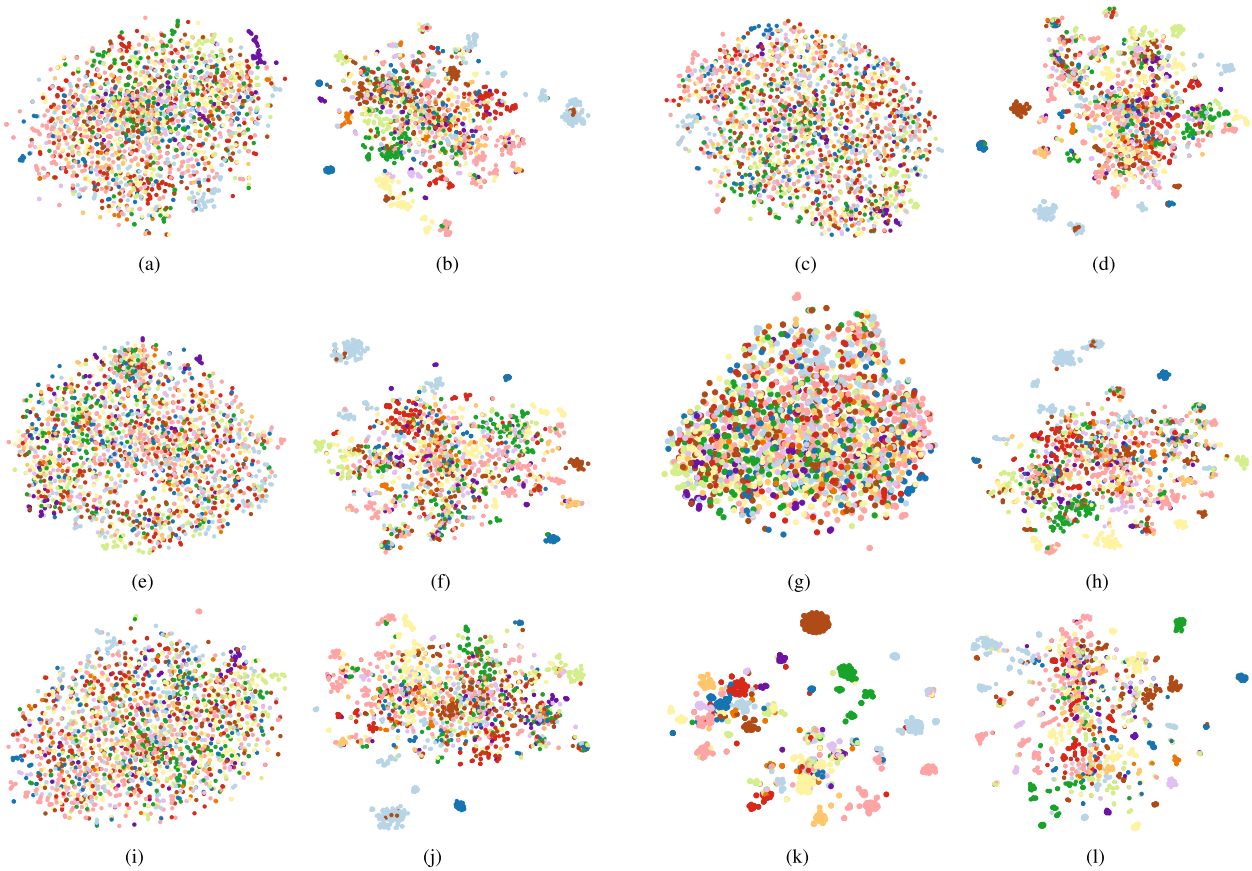

Fig. 3 *t*-SNE Scatterplots of Medical Codes Learned by Predictive Models. **a** MLP. **b** MLP+. **c** RNN. **d** RNN+. **e** RNN_a. **f** RNN_a+. **g** Dipole. **h** Dipole+. **i** RETAIN. **j** RETAIN+. **k** GRAM. **l** GRAM+

are significantly important to diagnosis prediction task, especially when the EHR data are insufficient.

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. The proposed framework includes two basic components: diagnosis code embedding and predictive model. In the diagnosis code embedding component, medical representations of diagnosis codes are learned from their descriptions with a convolutional neural network on top of pre-trained word embeddings. Based on the learned embeddings, the input visit information is embedded into a visit-level vector representation, which is then fed into the predictive model component. In the predictive model component, all the state-of-the-art diagnosis prediction models are redesigned to significantly improve the predictive performance by considering diagnosis code meanings. Experimental results on two real world medical datasets prove the effectiveness and robustness of the proposed framework for diagnosis prediction task. An experiment is designed to illustrate that the enhanced diagnosis prediction approaches outperform all the corresponding state-of-the-art approaches under insufficient EHR data. Finally, the learned medical code representations are visualized to demonstrate the interpretability of the proposed framework.

Abbreviations

BRNN Bidirectional recurrent neural network; CCS: Clinical classifications software; CNN: Convolutional neural networks; DAG: Directed acyclic graph; Dipole: Attention-based bidirectional recurrent neural networks; Dipole+: Enhanced attention-based bidirectional recurrent neural networks; EHR: Electronic health records; GRAM: Graph-based Attention model; GRAM+: Enhanced graph-based attention model; GRU: Gated recurrent unit; LSTM: Long-short term memory; MIMIC-III: Medical information mart for intensive care; MLP: Multilayer perceptron; MLP+: Enhanced multilayer perceptron; RETAIN: Reverse time attention mechanism; RETAIN+: Enhanced reverse time attention mechanism; RNN: Recurrent neural networks; RNN+: Enhanced recurrent neural network; RNN_a: Attention-based recurrent neural network; RNN_a+: Enhanced attention-based recurrent neural network; SDA: Stacked denoising autoencoders; T-LSTM: Time-aware long-short term memory

Acknowledgements

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. A short

version of this work titled "A General Framework for Diagnosis Prediction via Incorporating Medical Code Descriptions" was presented at the International Conference on Bioinformatics and Biomedicine in Madrid on 3-6 December 2018.

About this supplement
This article has been published as part of *BMC Medical informatics and Decision Making Volume 19 Supplement 6, 2019: Selected articles from the IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2018: medical informatics and decision making*. The full contents of the supplement are available online at https://bmcmedinformdecismak.biomedcentral.com/articles/supplements/volume-19-supplement-6.

Authors' contributions
FM, RC, JZ and JG developed the study concept and designed the model. RC acquired the EHR data, and FM processed the EHR data. FM, YW, HX and YY acquired and processed the medical code descriptions. FM, YW and YY programmed the CNN algorithm. FM carried out the experiments. FM, HX and JG analyzed the data and the experimental results. FM, RC, JZ and JG drafted the manuscript. All authors were involved in the revision of the manuscript. All authors read and approved the final manuscript.

Funding
Publication costs were funded in part by the US National Science Foundation under grant IIS-1747614.

Availability of data and materials
The MIMIC-III dataset can be obtained from the line: https://mimic.physionet.org/gettingstarted/access/.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Pennsylvania State University, State College, PA, USA. [2]University at Buffalo, Buffalo, NY, USA. [3]Georgia State University, Atlanta, GA, USA. [4]Beijing University of Technology, Beijing, China. [5]Kira Systems, Toronto, ON, Canada. [6]eHealth Inc., Mountain View, CA, USA.

Published: 19 December 2019

References

1. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: Review, opportunities and challenges. Brief Bioinform. 2017;19(6):1236–46.
2. Ma F, Chitta R, Zhou J, You Q, Sun T, Gao J. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: KDD. New York: ACM; 2017. p. 1903–11.
3. Choi E, Bahadori MT, Song L, Stewart WF, Sun J. Gram: Graph-based attention model for healthcare representation learning. In: KDD. New York: ACM; 2017. p. 787–95.
4. Choi E, Bahadori MT, Sun J, Kulas J, Schuetz A, Stewart W. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In: NIPS. Curran Associates, Inc.; 2016. p. 3504–12.
5. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606. 2016.
6. Dua S, Acharya UR, Dua P. Machine Learning in Healthcare Informatics vol. 56; 2014.
7. Suo Q, Zhong W, Ma F, Yuan Y, Huai M, Zhang A. Multi-task sparse metric learning for monitoring patient similarity progression. In: ICDM. IEEE; 2018. p. 477–86.
8. Ma F, Meng C, Xiao H, Li Q, Gao J, Su L, Zhang A. Unsupervised discovery of drug side-effects from heterogeneous data sources. In: KDD. New York: ACM; 2017. p. 967–76.
9. Yuan Y, Xun G, Ma F, Wang Y, Du N, Jia K, Su L, Zhang A. Muvan: A multi-view attention network for multivariate temporal data. In: ICDM. IEEE; 2018. p. 717–26.
10. Ma F, Wang Y, Xiao H, Yuan Y, Chitta R, Zhou J, Gao J. A general framework for diagnosis prediction via incorporating medical code descriptions. In: BIBM. IEEE; 2018. p. 1070–5.
11. Zhang S, Xie P, Wang D, Xing EP. Medical diagnosis from laboratory tests by combining generative and discriminative learning. arXiv preprint arXiv:1711.04329. 2017.
12. Zhang Y, Chen R, Tang J, Stewart WF, Sun J. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. In: KDD. New York: ACM; 2017. p. 1315–24.
13. Yuan Y, Xun G, Ma F, Suo Q, Xue H, Jia K, Zhang A. A novel channel-aware attention framework for multi-channel eeg seizure detection via multi-view deep learning. In: BHI. IEEE; 2018. p. 206–9.
14. Che Z, Kale D, Li W, Bahadori MT, Liu Y. Deep computational phenotyping. In: KDD. New York: ACM; 2015. p. 507–16.
15. Nguyen P, Tran T, Wickramasinghe N, Venkatesh S. Deepr: A convolutional net for medical records. IEEE J Biomed Health Inform. 2017 Jan;21(1):22–30.
16. Yuan Y, Jia K, Ma F, Xun G, Wang Y, Su L, Zhang A. Multivariate sleep stage classification using hybrid self-attentive deep learning networks. In: BIBM. IEEE; 2018. p. 963–8.
17. Suo Q, Ma F, Yuan Y, Huai M, Zhong W, Zhang A. Personalized disease prediction using a cnn-based similarity learning method. In: BIBM. IEEE; 2017. p. 811–6.
18. Suo Q, Ma F, Yuan Y, Huai M, Zhong W, Gao J, Zhang A. Deep patient similarity learning for personalized healthcare. IEEE Trans NanoBioscience. 2018;17(3):.
19. Cheng Y, Wang F, Zhang P, Hu J. Risk prediction with electronic health records: A deep learning approach. In: SDM. SIAM; 2016. p. 432–40.
20. Che Z, Cheng Y, Zhai S, Sun Z, Liu Y. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In: ICDM. IEEE; 2017. p. 787–92.
21. Ma F, Jing G, Suo Q, You Q, Zhou J, Zhang A. Risk prediction on electronic health records with prior medical knowledge. In: KDD. New York: ACM; 2018. p. 1910–19.
22. Pham T, Tran T, Phung D, Venkatesh S. Deepcare: A deep dynamic memory model for predictive medicine. In: PAKDD. Springer; 2016. p. 30–41.
23. Che C, Xiao C, Liang J, Jin B, Zho J, Wang F. An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease. In: SDM. SIAM; 2017. p. 198–206.
24. Che Z, Purushotham S, Cho K, Sontag D, Liu Y. Recurrent neural networks for multivariate time series with missing values. arXiv preprint arXiv:1606.01865. 2016.
25. Lipton ZC, Kale DC, Wetzel R. Modeling missing data in clinical time series with rnns. In: MLH; 2016. p. 253–70.
26. Lipton ZC, Kale DC, Elkan C, Wetzell R. Learning to diagnose with lstm recurrent neural networks. In: ICLR; 2015.
27. Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor ai: Predicting clinical events via recurrent neural networks. In: MLH; 2016. p. 301–18.
28. Choi E, Bahadori MT, Searles E, Coffey C, Thompson M, Bost J, Tejedor-Sojo J, Sun J. Multi-layer representation learning for medical concepts. In: KDD. New York: ACM; 2016. p. 1495–504.
29. Ma F, You Q, Xiao H, Chitta R, Zhou J, Gao J. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In: CIKM. New York: ACM; 2018. p. 743–52.
30. Suo Q, Ma F, Canino G, Gao J, Zhang A, Veltri P, Gnasso A. A multi-task framework for monitoring health conditions via attention-based recurrent neural networks. In: AMIA. American Medical Informatics Association; 2017. p. 1665–74.
31. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
32. Baytas IM, Xiao C, Zhang X, Wang F, Jain AK, Zhou J. Patient subtyping via time-aware lstm networks. In: KDD. New York: ACM; 2017. p. 65–74.
33. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259. 2014.
34. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.

35. Blunsom P, Grefenstette E, Kalchbrenner N. A convolutional neural network for modelling sentences. In: ACL. Association for Computational Linguistics; 2014. p. 655–65.
36. Kim Y. Convolutional neural networks for sentence classification. In: EMNLP. Association for Computational Linguistics; 2014. p. 1746–51.
37. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res. 2011;12(Aug):2493–537.
38. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Sig Process. 1997;45(11):2673–81.
39. Johnson AE, Pollard TJ, Shen L, Li-wei HL, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. Mimic-iii, a freely accessible critical care database. Sci Data. 2016;3:160035.
40. Team TTD. Theano: A python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688. 2016.
41. Zeiler MD. Adadelta: An adaptive learning rate method. arXiv preprint arXiv:1212.5701. 2012.
42. Maaten Lvd, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9(Nov):2579–605.

Publisher's Note