EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation

Xiancheng Xie
Shanghai Key Laboratory of Data Science, Shanghai Institute for Advanced Communication and Data Science, School of Computer Science, Fudan University, Shanghai, China
xcxie17@fudan.edu.cn

Yun Xiong
Shanghai Key Laboratory of Data Science, Shanghai Institute for Advanced Communication and Data Science, School of Computer Science, Fudan University, Shanghai, China
yunx@fudan.edu.cn

Philip S Yu
Computer Science Department, University of Illinois at Chicago, Chicago, IL 60607
psyu@uic.edu

Yangyong Zhu
Shanghai Key Laboratory of Data Science, Shanghai Institute for Advanced Communication and Data Science, School of Computer Science, Fudan University, Shanghai, China
yyzhu@fudan.edu.cn

ABSTRACT

Assigning standard medical codes (e.g., ICD-9-CM) representing diagnoses or procedures to electronic health record (EHR) is an important task in the medical domain. However, automatic coding is difficult since the clinical note is composed of multiple long and heterogeneous textual narratives (e.g., discharge diagnosis, pathology reports, surgical procedure notes). Furthermore, the code label space is large and the label distribution is extremely unbalanced. The state-of-the-art methods mainly regard EHR coding as a multi-label text classification task and use shallow convolution neural network with fixed window size, which is incapable of learning variable *n*-gram features and the ontology structure between codes. In this paper, we leverage a densely connected convolutional neural network which is able to produce variable *n*-gram features for clinical note feature learning. We also incorporate a multi-scale feature attention to adaptively select multi-scale features since the most informative *n*-grams in clinical notes for each word can vary in length according to the neighborhood. Furthermore, we leverage graph convolutional neural network to capture both the hierarchical relationships among medical codes and the semantics of each code. Finally, We validate our method on the public dataset, and the evaluation results indicate that our method can significantly outperform other state-of-the-art models.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; Semantic networks; • **Applied computing** → *Health informatics*;

KEYWORDS

EHR Coding, Densely Connected CNN, Multi-scale Feature Attention, Graph Convolutional Neural Network

ACM Reference Format:
Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357897

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357897

1 INTRODUCTION

Electronic health record (EHR) coding is a standard procedure of extracting diagnosis and procedure codes from clinical notes pertaining to a patient's visit. The clinical note is mostly composed of multiple long and heterogeneous textual narratives (e.g., discharge diagnosis, pathology reports, surgical procedure, past illness history) authored by different healthcare professionals. The diagnosis and procedure codes used in the EHR coding are from the International Classification of Diseases (ICD), which provides a hierarchy of diagnosis or procedure codes of diseases, disorders, injuries, signs, symptoms, etc. Fig. 1 shows a code assignment example.

Manual coding is error-prone, time-consuming and tedious for human coders, which requires the coders having a thorough knowledge of ICD and following a complex set of guidelines. The cost incurred by coding errors can be expensive. For example, if a hospital coder indicates the diagnosis code for "left heart failure" (code 428.1) instead of the one for "acute systolic heart failure" (code 428.21), the difference could mean thousands of dollars for billing [1]. Despite automatic EHR coding has been studied since at least the 1990s [3], it still presents several challenges: (1) clinical notes always include multiple long textual narratives (more than 1500 tokens on average), however, only a small part of tokens are relevant for specific ICD codes, which seems to seek a needle in the haystack. Furthermore, there are many abbreviations and symptoms in clinical notes, which cause ambiguity and misunderstanding for ICD codes matching; (2) the code label space is very high-dimensional,

[1] https://nyti.ms/2oxrjCv

Discharge Summary

History of Present Illness:
This is a 54-year-old gentleman with a history of deep venous thrombosis and pulmonary embolism in [**2155**], history of hypertension, and atrial fibrillation who presented to the Emergency Department with extreme dyspnea on exertion and weakness.
.....
Major Surgical or Invasive Procedure:
Attempt Repair of Ruptured Aortic Aneurysm
.....
Brief Hospital Course:
The patient was brought to the cardiac catheterization laboratory and a right heart catheterization revealed a pulmonary artery pressure of 49/17, right ventricular pressure
.....
PERTINENT RADIOLOGY/IMAGING:
Electrocardiogram revealed sinus tachycardia with a rate of 120. Intervals were otherwise normal. He had poor R wave progression. No ST changes.
.....
Discharge Diagnosis:
1. Pulmonary embolism; presenting as cardiogenic shock. 2. Atrial fibrillation/atrial flutter. 3. Renal insufficiency.
.....

ICD-9-CM Codes:
37.21 38.7 88.52 88.56 89.8 96.04 99.10 99.62 96.71 270.4 403.91 415.19 427.32 428.0 518.81 780.57

Figure 1: A exemplar clinical note from MIMIC-III [5]. The clinical note consists of multiple heterogeneous long narratives with more than 1500 tokens on average.

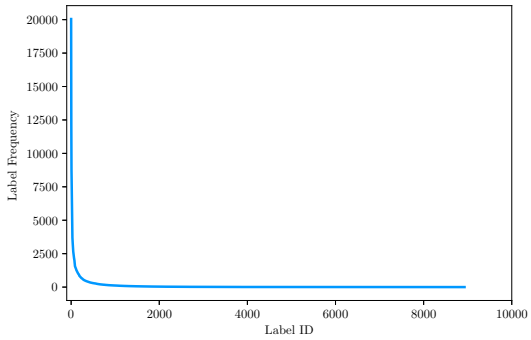


Figure 2: The ICD-9-CM label distribution for MIMIC-III [5] dataset. There are about 5411 of all the 8929 labels occurring less than 10 times.

with over 18000 codes in ICD-9-CM [2], and over 170000 codes for ICD-10 [3]. Moreover, the label distribution is extremely unbalanced — most of the codes appear very seldom while a few codes occur several orders of magnitude more than others; As shown in Fig. 2, there are about 5411 of all the 8929 labels occurring less than 10 times in the MIMIC-III [5] dataset; (3) a clinical note may indicate a large number of diagnosis and procedure codes (16 labels on average per instance for MIMIC-III [5]). In many cases, the difference between diagnosis or procedure subtypes is very subtle. It is common for inexperienced medical coders to incorrectly select subtypes. Hence, there is a need for machine learning approaches which are able to extract more informative features from long clinical notes and are generally more robust to data sparsity.

The current state-of-the-art methods [12, 22] use Text-CNN [7] to extract document representation. For better adapting to the multi-label setting, they employ a label-dependent attention mechanism, which allows their models to learn distinct document representations for individual label. However, these models employ convolution filters of fixed size, which is equivalent to extracting fixed-size n-gram features. There are apparent drawbacks for multi-label clinical classification which needs to have variable-size features such as phrases about diseases or procedures for better representation. For example, assigning ICD codes for sentence "*His admission chest x-ray demonstrated severe prominence of the right main pulmonary artery*" requires to extract bigram features *"severe prominence"* and 4-gram features *"right main pulmonary artery"*, where the first phrase describes the disease and severity and the latter describes the disease location. However, when applying filters of size 1, we not only seperate these adjectives from their objects (e.g., severe / prominence, right / main pulmonary artery), but also seperate the location phrase (e.g., main / pulmonary / artery). The former seperation may cause incorrect disease subtypes assignment while the latter may cause unrelevant disease code assignment (e.g., artery occlusion). Current methods [12, 22] tend to use large filter size (e.g., 10) as a compromise. However, using filters with large sizes will introduce irrelevant information, and thus the response for the bigram *"severe prominence"* will undesirably decrease. An ordinary solution for variable-size n-gram features extraction is to leverage filters with different kernel sizes. However, such a solution is equivalent to learn several disconnected networks in parallel for ensemble. Thus, the interaction of feature maps from different filter sizes are not fully exploited, which makes the model difficult to train and the precision hard to achieve. Moreover, the direct concatenation over feature maps from different filter sizes will decrease the response of the most informative phrase, which makes the model give inaccurate prediction.

Based on the above analysis, we build dense connections between upstream and downstream convolutional layers, which encourages the better reuse of downstream features. Take sentence *"severe prominence of the right main pulmonary artery"* as an example, the 4-gram features for "right main pulmonary artery" can reuse 1-gram features *"right"* and trigram features "main pulmonary artery" rather than two bigram features *"right main"* and *"pulmonary artery"*. Instead of direct concatenation over feature maps from different scales, we propose a multi-scale feature attention to adaptively select most informative n-gram features for each word according to its neighbor context. Our proposed multi-scale feature attention can effectively merge feature maps from different filter sizes without decreasing the response of the most informative phrase resulting in a huge improvement over state-of-the-art methods [12, 22]. On the other hand, each ICD code has an official description that tells the semantics of the code and there is a hierarchical structure among the ICD codes. For example, "proximal pancreatectomy (52.51)", "distal pancreatectomy (52.52)" and "radical subtotal pancreatectomy (52.53)" are all children of "partial pancreatectomy (52.5)". As described above, the label distribution is extremely unbalanced. Generally, nodes close to one another in medical ontologies ought to be associated with similar semantic embeddings, allowing us to transfer knowledge among them. Therefore, proper use of medical ontologies will be helpful when we lack enough data for the nodes in the ontology to train deep learning models. Based on above consideration, we leverage graph convolutional neural network [9] to capture the hierarchical relationships among medical codes and

[2]ICD-9-CM is a version of ICD-9 used in the United States.
[3]https://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm

the semantics of each code. Overall, our major contributions can be summarized as follows:

- We utilize a densely connected convolutional neural network for clinical note encoding which can produce variable n-gram features layer by layer.
- We incorporate multi-scale feature attention to adaptively select most informative n-gram features.
- We leverage graph convolutional neural network to capture the hierarchical relationships among medical codes and the semantics of each code.
- Following prior works we quantitatively evaluate our approach on real EHR datasets, demonstrating the effectiveness of our proposed method.

2 RELATED WORK

EHR coding is a hot research topic in the medical domain and has attracted many attentions. Many recent approaches regard EHR coding as a multi-label text classification. Perotte et al. [13] ultilized "flat" and "hierarchical" SVMs based on tf-idf document features; the former treats each code as an individual prediction, while the latter exploits ICD code ontology for hierarchical classification. Kavuluru et al. [6] evaluated supervised learning approaches (e.g, SVM, naïve Bayes, and logistic regression) and performed feature selection over three private datasets. Scheurwegs et al. [16] incorporated structured and unstructured text information from EHRs and used a feature selection approach.

Recent advances in neural networks have also been put to use for EHR coding. Shi et al. [18] utilized character-aware LSTMs to generate hidden representations of written diagnosis descriptions and ICD codes, and designed an attention mechanism to address the mismatch between the descriptions and corresponding codes. Prakash et al. [14] used dense memory networks that draw from clinical notes as well as Wikipedia, to predict most frequent 50 and 100 codes. Another recent neural architecture is the Grounded Recurrent Neural Network [21], which employs a modified GRU with dimensions dedicated to predicting the presence of individual labels. Baumel et al. [1] applied a hierarchical GRU model with label-dependent attention layer to classify codes while providing an explainable decision process. The current state-of-the-art method [12] exploits text convolutional neural network [7] for document encoding and introduces a label-dependent attention mechanism to learn distinct document representations for the individual label. Wang et al. [22] proposed a label-word joint embedding model, which embeds label and word into the same space and exploits the cosine similarity between them for predicting the labels.

Except for using neural networks for better document feature learning, there are some works focus on code label ontology, code sparsity, and external resources. Wang et al. [23] introduced a pairwise regularization term to capture the disease code relevance under multi-label setting. Xie et al. [24] leveraged Tree-LSTM [20] to capture code structures and regard EHR coding as a semantic matching problem. Rios et.al [15] regarded clinical coding as a few-shot learning problem and introduced matching network [19] for extreme label imbalance. In addition, there are many methods [17, 25, 27] using extra information (e.g., patient tabular data, PubMed text corpus, and laboratory measurements, etc.) for multi-modal EHR coding.

Table 1: Important Notations.

Symbol	Definition
$\mathbf{L}$	the set of ICD-9-CM code
$\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m$	the word sequences of clinical note
$\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{n_l}$	the description sequences of l-th code
$\boldsymbol{X}$	the word embeddings of clinical note
$\boldsymbol{X}_k$	the outputs of k-th convolution block
$\boldsymbol{X}^{scale}$	features after multi-scale attention
$\boldsymbol{X}^{final}$	features after label-dependent attention
$\boldsymbol{V}$	the representation of code description
$\boldsymbol{V}^k$	the feature outputs of k-th layer of graph propagation module
$\boldsymbol{V}^{cls}$	code classifiers after graph propagation

However, in this paper, we only use free-text discharge summaries for EHR coding following prior works [12, 21, 22, 24, 26].

Our approach is most similar to [12], as in their study, we use CNN layers with label-dependent attention. However, our approach has following major novel improvements. Firstly, we use densely connected convolutional layers which encourages better downstream feature reuse to extract variable n-gram features. Secondly, we exploit multi-scale attention to adaptively select most informative n-gram features according to the word neighborhood. Thirdly, we use graph convolutional neural network [9] to capture both the hierarchical relationships among medical codes and the semantics of each code. The quantitative and qualitative results on real EHR dataset demonstrate the effectiveness of our proposed method.

3 METHODS

As mentioned before, the clinical note consists of multiple heterogeneous long narratives, and the code label distribution is extremely unbalanced. In the following, we develop a model with multi-scale feature attention and structured knowledge graph propagation (MSATT-KG), which takes into consideration these properties. We first introduce an overview of our proposed method, and then a detailed description of each component.

3.1 Overview

An overall pipeline of our proposed method is visualized in Fig 3. Similar to prior work [12], we regard EHR coding as a multi-label document classification problem. Let $\mathbf{L}$ represents the set of ICD-9-CM codes. Then EHR coding for document aims to determine $y_l \in \{0, 1\}$ for all $l \in \mathbf{L}$. The clinical document can be represented as m word sequences $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m\}$. In addition, each code $l \in \mathbf{L}$ has an offical description which can be represented as n_l words $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{n_l}\}$. Our method is mainly composed of three parts: (1) clinical document multi-scale feature extraction; (2) two-level attention mechanism for better document representation learning; (3) structured knowledge graph propagation. Firstly, we build our model with an embedding layer stacked with densely

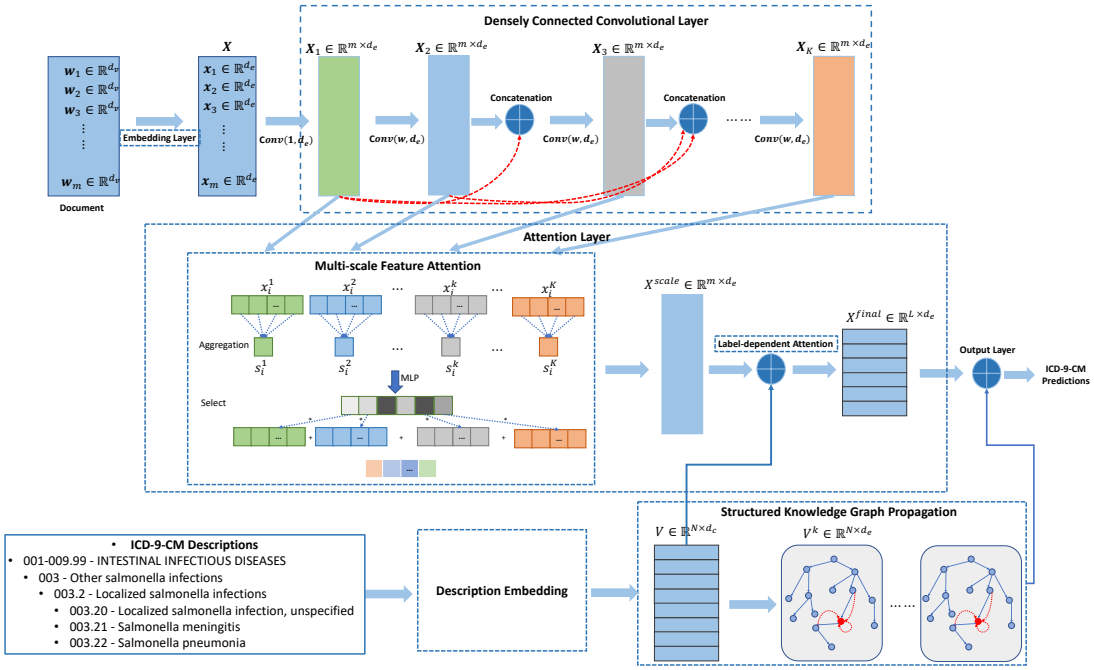


Figure 3: An overall pipeline of our proposed model.

connected convolution blocks to generate multi-scale features since densely connections encourage better downstream feature reuse. After multi-scale feature extraction, we firstly use a multi-scale attention mechanism over different scales to adaptively select most informative n-gram features for each word according to the neighborhood. Rather than aggregating with a pooling operation, we then apply a label-dependent attention mechanism over words to select the most relevant words of the document for each possible code for overcoming the situation of needle in a haystack. Finally, we use proposed structured knowledge graph propagation module to tranfer knowledge among codes to overcome extremely imbalanced label distribution. The important notations used in our method are summaried in Table 1. We now describe each of these elements in more detail.

3.2 Embedding Layer

Each word $\boldsymbol{w} \in \mathbb{R}^{d_v}$ is mapped to $\boldsymbol{x} \in \mathbb{R}^{d_e}$ using the embedding weight $W_e \in \mathbb{R}^{d_v \times d_e}$, where d_v is the total vocabulary size and d_e is the embedding size. Thus each clinical note with m words can be embeded as a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m]_{m \times d_e}$. Following prior work [12], we pre-train the embedding layer on all the text in training set using the continuous bag-of-words (CBOW) word2vec[4] [11] using gensim [5], with an embedding size of 100, window size of 5, no minimum count for 5 epochs.

[4]We use word2vec for fair comparision with prior works, actually we can use any other word embedding methods including BERT[4].
[5]https://radimrehurek.com/gensim

3.3 Densely Connected Convolutional Layer

Given clinical note embeddings $[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, our CNN consists of K stacked convolution blocks via densely connections. That is, let $\boldsymbol{X}_k = [\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, \cdots, \boldsymbol{x}_m^k]_{m \times d_c}$ represent the outputs of k-th intermediate convolution block, where $1 \leq k \leq K$, and d_c is the dimension of the transformed feature representation. The convolution block takes as input the outputs of all downstream layers $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{k-1}$, and generates intermediate feature representations as below:

$$\boldsymbol{X}_k = \mathbb{F}(\boldsymbol{W}_k, [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{k-1}]) \tag{1}$$

where $[\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_{k-1}]$ refers to the concatenation of the feature maps of layers $1, 2, \cdots, k-1$ and each $X_i \in \mathbb{R}^{m \times d_c}$, $\mathbb{F}$ refers to the convolution function with filter W_k. The weight matrix W_k $(2 \leq k \leq K)$ consists of $(k-1) \times d_c$ filters, each of which is of size $w \times d_c$, where w is the kernel size (e.g. $w = 3$). Notice that the weight matrix W_1 for the first layer has d_c filters with size $1 \times d_e$ since the first layer takes as input the clinical embeddings $\boldsymbol{X}$. To preserve the input length, we perform zero-padding on the two sides of input for every layer. Note that X_k represents a specific n-gram features for clinical document. More specifically, without loss of generality, taking $w = 3$ as an example, the first layer's receptive field is 3 corresponding to trigram features, the second layer corresponding to 5-gram features, the third layer corresponding to 7-gram features, and so on. Compared to tranditional sequentially stacked convolutional blocks, densely connected convolution computes upstream features (larger-scale n-grams) by considering downstream features. Thus smaller n-grams will be fully used to get larger n-grams, resulting in the flexibility of extracting multi-scale features.

3.4 Attention Layer

Our attention layer consists of two components including multi-scale feature attention and label-dependent attention. Through above densely connected convolution layer, we can get multi-scale features representing as $[\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_K]_{K \times m \times d_c}$. We firstly adaptively selects most informative scale features at each position of a text through our multi-scale feature attention mechanism. Then we attend to most relevant parts of input for each code using label-dependent attention mechanism. We now describe these two components in more detail.

3.4.1 Multi-scale Feature Attention. Given multi-scale features, our attention mechanism aims to select features of different scales at each position of document according to neighborhood words. As shown in Fig 3, at each position i, the attention mechanism computes a weight distribution over 1-th to K-th layer features representing as $\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \cdots, \boldsymbol{x}_i^K$. The multi-scale feature attention consists of two steps including ***aggregation*** and ***select***. We firstly develop a scalar value s_i^k to represent descriptor of each scale $\boldsymbol{x}_i^k$ at position i as follows:

$$s_i^k = \sum_{t=1}^{d_c} \boldsymbol{x}_i^k(t) \tag{2}$$

The scalar s_i^k aggregates the response values of $\boldsymbol{x}_i^k$. The scalar s_i^k can be used as a descriptor of $\boldsymbol{x}_i^k$, since these values of $\boldsymbol{x}_i^k$ indicate the responses of multi-scale n-grams and they are yielded by applying d_c filters on the preceding feature maps. After obtaining aggregated scalar s_i^k, we compute the attention weights to select most informative scale features as below:

$$\begin{aligned} \boldsymbol{\alpha}_i &= softmax(\mathbb{F}_{mlp}(\boldsymbol{s}_i)) \\ \boldsymbol{s}_i &= [s_i^1, s_i^2, \cdots, s_i^K] \\ \boldsymbol{\alpha}_i &= [\alpha_i^1, \alpha_i^2, \cdots, \alpha_i^K] \end{aligned} \tag{3}$$

where $\mathbb{F}_{mlp}$ is a multi-layer perceptron. Then we use attention weights $\boldsymbol{\alpha}_i$ to get final scale-aware features at each position i as follows:

$$\boldsymbol{x}_i^{scale} = \sum_{k=1}^{K} \alpha_i^k \boldsymbol{x}_i^k \in \mathbb{R}^{d_c} \tag{4}$$

After our multi-scale feature attention module, we get final representation:

$$\boldsymbol{X}^{scale} = [\boldsymbol{x}_1^{scale}, \boldsymbol{x}_2^{scale}, \cdots, \boldsymbol{x}_m^{scale}] \in \mathbb{R}^{m \times d_c}$$

which will be fed into next attention layer.

3.4.2 Label-dependent Attention. It is typical to apply pooling (e.g., max-pooling or mean-pooling) over words to select maximum or average value at each dimension for text classification [7]. However, EHR coding aims to assign multiple labels for each document, and different code should focus on different parts of words. Based on above analysis, we apply a label-dependent attention over words to select most relevant phrase for specific code. Formally, for each label l, we compute the bilinear pooling product as follows:

$$\boldsymbol{\alpha}_l = softmax(\boldsymbol{X}^{scale} \boldsymbol{v}_l) \tag{5}$$

where $\boldsymbol{v}_l \in \mathbb{R}^{d_c}$ is the description vector for l-th label. The attention $\boldsymbol{\alpha}_l$ is then used to compute vector representation for each label,

$$\begin{aligned} \boldsymbol{x}_l^{final} &= \sum_{i=1}^{m} \alpha_l^i \boldsymbol{x}_i^{scale} \\ \boldsymbol{\alpha}_l &= [\alpha_l^1, \alpha_l^2, \cdots, \alpha_l^m] \\ \boldsymbol{X}^{final} &= [\boldsymbol{x}_1^{final}, \boldsymbol{x}_2^{final}, \cdots, \boldsymbol{x}_{\mathbf{L}}^{final}] \end{aligned} \tag{6}$$

3.5 Structured Knowledge Graph Propagation

Our proposed structured knowledge graph propagation aims to learn code classifiers which takes code hierarchical structures and correlations into consideration. As mentioned earlier, there are hierarchical structures among ICD codes and the l-th code has an official description consisting of n_l words $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_{n_l}\}$. In order to get initial node features, we firstly feed every code description into our description embedding module. More specifically, our description embedding module is similar to Text-CNN [7] including embedding layer, convolutional layer, and pooling layer. Note that, we also pre-train embedding layer using full clinical notes of training set. After convolution block, we use max-pooling over the words to get final code representation as $\boldsymbol{V} \in \mathbb{R}^{N \times d_c}$, where N is the size of all ICD-9-CM codes (including parent nodes). Different from [12] using code representation $\boldsymbol{V}$ as regularization, we directly use $\boldsymbol{V}$ to select most relevant phrase in label-dependent attention. As mentioned earlier, the label distribution is extremely unbalanced, and thus it is difficult to learn good classifiers for these labels which only have few samples. Thus we leverage graph convolutional neural network to capture code relationships and correlations for learning better code classifiers. That is, given a structured graph with N nodes and with d_c input features per node, every node update information by combining the information from its children and parents every step. That is, we employ following propagation rule for i-th node to perform convolutions on the k-th layer graph:

$$\begin{aligned} \boldsymbol{v}_i^k &= \mathbb{F}(\boldsymbol{W}_{self}^k \boldsymbol{v}_i^{k-1} + \sum_{j \in \mathbb{N}_p} \frac{\boldsymbol{W}_p^k \boldsymbol{v}_j^{k-1}}{|\mathbb{N}_p|} + \sum_{j \in \mathbb{N}_c} \frac{\boldsymbol{W}_c^k \boldsymbol{v}_j^{k-1}}{|\mathbb{N}_c|}) \\ \boldsymbol{V}^k &= [\boldsymbol{v}_1^k, \boldsymbol{v}_2^k, \cdots, \boldsymbol{v}_N^k] \end{aligned} \tag{7}$$

where $\boldsymbol{V}^0 = \boldsymbol{V}$, $\mathbb{F}$ is the activation function, $\boldsymbol{W}_{self}^k$, $\boldsymbol{W}_p^k$, $\boldsymbol{W}_c^k$ are parameter matrix, $\mathbb{N}_p, \mathbb{N}_c$ are the index set of the j-th label's parents and children respectively. After D layer above graph-based propagation, we get the final node features $\boldsymbol{V}^D = [\boldsymbol{v}_1^D, \boldsymbol{v}_2^D, \cdots, \boldsymbol{v}_N^D] \in \mathbb{R}^{N \times d_e}$, and then we extract leaf node features from $\boldsymbol{V}^D$ referring to as $\boldsymbol{V}^{cls} \in \mathbb{R}^{\mathbf{L} \times d_e}$ as code classifiers, which is used for final multi-label classification.

3.6 Output Layer

Given the label-wise document representation from attention layer which is represented as $[\boldsymbol{x}_1^{final}, \boldsymbol{x}_2^{final}, \cdots, \boldsymbol{x}_{\mathbf{L}}^{final}]$,the output layer use a linear layer following a sigmoid transformation for each class, that is,

$$p_l = sigmoid(\boldsymbol{V}_l^{cls} \boldsymbol{x}_l^{final} + b_l)\ l \in \{1, 2, \cdots, \mathbf{L}\} \tag{8}$$

where $\boldsymbol{V}_l^{cls}$ is l-th classifier, p_l is the probability of l-th class. The most oridary method to convert p_l into label predictions is to simply threshold at 0.5. Note that since even the most frequent class in the training examples occurs in only 37%, the network is strongly biased towards negative predictions. A simple way to solve this unbalanced problem is to optimize the threshold value for each label. However, searching for the optimal threshold of each label is computational expensive in large label spaces. Here, we use a simple meta-learner to evaluate the number of labels $\hat{h}$ that the document should be annotated with. More specifically, we use following regression output layer:

$$\hat{h} = ReLU(\boldsymbol{W}^{reg} g(\boldsymbol{X}^{scale}) + b_h) \tag{9}$$

where $\boldsymbol{W}^{reg}$ is the parameter matrixs, $g(\boldsymbol{X}^{scale})$ means max-pooling over words at $\boldsymbol{X}^{scale}$ which can be seen as a global represent for the document. At training time, we optimize the classification layer defined in Eq.8 with binary-cross-entropy loss, and the regression layer defined in Eq.9 with mean-square-error. that is, the total loss function can be defined as follows:

$$loss = l_{cls} + \lambda l_{reg} \tag{10}$$

where λ specifies the weight of the regression loss. At test time, we rank each label by its score p_l . Next, $\hat{h}$ is rounded to the nearest integer and we predict the top $\hat{h}$ ranked labels.

4 EXPERIMENTS

4.1 Database

We evaluate our mothod and baseline methods on the publicly avaiable MIMIC-III dataset [5] for ICD-9-CM coding. Note that some prior methods report results on MIMIC-II dataset, which is an older version and a subset of MIMIC-III. Since the fact that MIMIC-III is more comprehensive and current, we only focus on the results of MIMIC-III. MIMIC-III dataset includes the electronic health record (EHR) of inpatient in the hospital intensive care unit (ICU). There are different event notes for each stay in the ICU, including discharge summary report, discharge summary addendum, nursing notes, radiology notes, etc. Following prior works [12–15, 22], we focus on discharge summaries, which summarize the information about a stay into a single document. We also concatenate the discharge summary addenda to discharge summary report as prior work [12].

In MIMIC-III, each admission is annotated by human coders with several ICD-9-CM codes. There are 8929 unique ICD-9-CM codes present in the datasets, including 6918 diagnosis codes and 2011 procedure codes. We use the train, validation, and test splits publicly shared by prior work [12]. These splits are done in patient level thus no patient appears in both the training and test sets. There are two experiment settings for comparison with prior works: 1) in full-label setting, we use all discharge summaries with 8,929 labels, resulting in 47,724 discharge summaries from 36,998 patients for training set, with 1,632 summaries and 3,372 summaries for validation and testing, respectively; 2) in top-50 label setting, we only predict 50 most frequent labels, and filter each dataset down to the instances that have at least one of the top 50 most frequent codes, resulting in 8,607 summaries for training, 1,574 for validation, and 1,730 for testing, respectively. Detailed statistics for the two settings are summarized in Table 3.

Table 2: Hyperparameter candidates used in Hyperband search, and final selected values.

settings	hyperparameters	candidates	selected
top-50 label	d_e	80-200	200
	K	3,4,5,6,7,8	6
	D	1,2,3,4,5	2
	λ	(1.0, 0.1, 0.01, 0.001)	0.1
full-label	d_e	80-150	100
	K	3,4,5,6	5
	D	1,2,3,4	2
	λ	(1.0, 0.1, 0.01, 0.001)	0.01

For preprocessing the document, we convert all tokens to lower case, remove non-alphabetic tokens, and replace tokens that appear in fewer than three times with special "UNK" token. This results in 51,867 unique words in two settings. Following prior work [12], we pre-train word embeddings on all the text in training set using the continuous bag-of-words (CBOW) word2vec [11] implemented by gensim, with an embedding size of 100, window size of 5, no minimum count for 5 epochs. All documents are truncated to a maximum length of 4000 tokens.

4.2 Implementation Details

We implement the proposed model on PyTorch[6] and train on 8× 1080Ti GPUs. For training settings, we use Adam [8] with learning rate of 0.0001 and $\beta_1 = 0.9$, $\beta_2 = 0.999$. The mini-batch size is set to 32, 16 for top-50 label setting, full-label setting respectively. We train all models with early stopping, using precision@8 on the validation set as stopping criterion with patience of 4 epochs. We use a dropout of 0.2 after the embedding layer. The convolutional kernel size is set to 3, and the weight decay is set to 0.00005. To optimize other hyperparameters, we use Hyperband [10] implemented by Ray[7] since it is faster than other Bayesian optimization. Hyperband requires setting a maximum training epochs for hyperparameter searching, we set it to 30, as well as an additional pruning parameter σ, which is set to 3. We choose to optimize following hyperparameters with Hyperband: the number of filters d_e, the number of convolution layers K, the number of structured graph propagation layers D, the weight λ. Table 2 shows the hyperparameters used for candidates and final selected value, using precision@8 or precision@5 value on the validation set as the maximum target.

4.3 Baseline Models

In order to demonstrate the effectiveness of our proposed method, we compare proposed MSATT-KG with the state-of-the-art methods for EHR coding.

Flat and hierarchical SVMs [13] use 10,000 tf-idf unigram features, they train 8,929 binary SVMs for flat SVMs and 11,693 SVMs for hierarchical SVMs. For flat SVMs, a code is considered present if its SVM give a positive output. For hierarchical SVMs, a code is

[6] https://pytorch.org/
[7] https://github.com/ray-project/ray

Table 3: Descriptive statistics for MIMIC-III dataset.

settings / split		samples	tokens	average tokens	codes	unique codes	average codes
full-label set	train	47,723	68,446,108	1,434	748,306	8,692	15.68
	test	3,372	5,837,889	1,731	60,666	4,085	17.99
	val	1,631	2,812,609	1,724	28,402	3,012	17.41
top-50 label set	train	8,066	11,921,617	1,478	45,927	50	5.69
	test	1,729	3,049,336	1,763	10,428	50	6.03
	val	1,573	2,735,514	1,739	9,241	50	5.87

predicted present only if the all nodes of the path from the root to this node give positive outputs.

Logistic regression is a bag-of-words logistic regression model.

Bi-GRU uses bidirectional gated recurrent unit for document representation learning.

Text-CNN [7] is a single layer convolutional neural network without label-dependent attention, we only use max-pooling over words to extract representation for all codes.

Selected feature [16] incorporates structured and unstructured text information from EHRs and uses a feature selection approach.

C-MemNN [14] imploys dense memory networks that draw from clinical notes as well as Wikipedia for extra resources, to predict most frequent 50 labels.

Attentive LSTM [18] utilizes character-aware LSTM to generate hidden representations of written diagnosis descriptions and ICD codes, and designs an attention mechanism to address the mismatch between the numbers of descriptions and corresponding codes. We compare MSATT-KG with it on top-50 label set since their method only evaluates on top-50 label set.

CAML and DR-CAML [12] have achieved state-of-the-art results on MIMIC-III dataset. CAML uses Text-CNN [7] for document representation learning. For overcoming the situation of needle in a haystack, they propose a label-dependet attention to learn most informative representation for every specific code. DR-CAML enhance CAML by label description regularization on the final classification weights. They hypothesize that a code's description should semantically similar to the segments of input text that highlight by their label-dependent attention. Thus they extract label description representation by Text-CNN architecture and then give a regularization between the code representation and final classification layer's weights using mean-square-loss. However, DR-CAML performs worse on most metrics than CAML on full-label set, so we compare our results with DR-CAML on top-50 label set, and CAML on full-label set. We run their model using their publicly available code [8].

LEAM [22] proposes a label-word joint embedding model which embeds label and word into the same space and exploits the cosine similarity between them for predicting the labels. We compare MSATT-KG with LEAM on top-50 label set.

4.4 Evaluation Metrics

For fairly comparing with prior works, we report a variety of metrics, including micro F1 score, macro F1 score, and area under the ROC curve (AUC). We also report the micro and macro F1 score for diagnosis and procedure codes for comparability with previous works [12, 13]. The micro F1 score is calculated by treating each (text, code) pair as a separate prediction while macro F1 score is calculated by averaging metrics computed per-label. The macro F1 score places much more emphasis on rare label prediction. We also report precision at n (e.g., p@5, p@15), which computes the fraction of the top-n highest scored labels that are present in the ground truth. We choose $n = 5$ and $n = 8$ following prior works [12, 22] on top-50 label set, full-label set respectively. For full-label setting, we also compute p@15, which roughly corresponds to the average number of codes in the dataset.

4.5 Results and Analysis

4.5.1 Quantitative Results. we report all quantitative metrics for both top-50 label set and full-label set on Table 5 and Table 4 respectively. We firstly analyse the results on full-label set since EHR coding under this setting is more difficult. Our proposed MSATT-KG gives the strongest results on all metrics due to our proposed multi-scale feature attention and structured knowledge graph propagation. More specifically, MSATT-KG outperforms state-of-the-art method CAML by a considerable margin, with **2.4** improvements on micro F1 score, **1.9** improvements on p@8, and **2.0** improvements on p@15 respectively. MSATT-KG also has a **1.5** improvements on Macro AUC, and **0.6** improvements on Micro AUC respectively. Furthermore, we observe that the logistic regression baseline is substantially worse than other neural architectures since it uses hand craft document features (Bow). Text-CNN is comparable to Bi-GRU since the sequential information of words is not as important as it is in language models [2] or sentiment analysis [26]. For EHR coding, the key words, phrases, and their composition are more important. CAML yields substantial improvements over vanilla Text-CNN due to their proposed lable-dependent attention. Our proposed multi-scale attention and structured graph propagation further improve the performance although the performace of CAML is relatively high. It is worth noting that hierarchy SVMs outperforms flat SVMs since it uses the hierarchy of the codes and utilizes the dependency betweeen codes. Hierarchy SVMs enhances the recall for the sparse classes as shown in [13], however, their performance is still lower than CAML and MSATT-KG since they use unigram tf-idf features, which have typical limitations such as no phrases, no locality, no word composition. In addition, Selected Feature [16] reports the results distinguished between diagnosis and procedure codes, which outperforms Text-CNN. However, their method utilizes the strcutured data for text representation, while

[8] https://github.com/jamesmullenbach/caml-mimic

Table 4: Results on MIMIC-III, full-label set.

model	AUC (%)		F1 (%)				*p*@*n*(%)	
	macro	micro	macro	micro	diag	proc	8	15
Logistic Regreesion	56.1	93.7	1.1	27.2	24.2	39.8	54.2	41.1
Selected Feature [16]	-	-	-	-	42.8	55.5	-	-
Bi-GRU	82.2	97.1	3.8	41.7	39.3	51.4	58.5	44.5
Flat SVMs [13]	-	-	-	39.7	38.5	42.1	-	-
Hierarchy SVMs [13]	-	-	-	44.1	43.3	46.0	-	-
Text-CNN [7]	80.6	96.9	4.2	41.9	40.2	49.1	58.1	44.3
DR-CAML [12]	89.7	98.5	8.6	52.9	51.5	59.5	69.0	54.8
CAML [12]	89.5	98.6	8.8	53.9	52.4	60.9	70.9	56.1
MSATT-KG	**91.0**	**99.2**	**9.0**	**55.3**	**54.0**	**62.3**	**72.8**	**58.1**

Table 5: Results on MIMIC-III, top-50 label set.

model	AUC (%)		F1 (%)		
	macro	micro	macro	micro	*p*@5
Logistic Regreesion	82.9	86.4	47.7	53.3	54.6
Bi-GRU	82.8	86.8	48.4	54.9	59.1
C-MemNN [14]	83.3	-	-	-	42.0
Attentive LSTM [18]	-	90.0	-	53.2	-
Text-CNN [7]	87.6	90.7	57.6	62.5	62.0
LEAM [22]	88.1	91.2	54.0	61.9	61.2
CAML [12]	87.5	90.9	53.2	61.4	60.9
DR-CAML [12]	88.4	91.6	57.6	63.3	61.8
MSATT-KG	**91.4**	**93.6**	**63.8**	**68.4**	**64.4**

MSATT-KG only uses free-text summaries. The large improvements between Selected Feature and logistic regreesion reveals that effective text representation is very important for EHR coding, and our proposed densely connected multi-scale convolution can effectively learn good text representation. To compare with prior works, we also evaluate the result on the top-50 label set. Under this setting, we can see strong improvements over all reported metrics against all baselines. More specifically, MSATT-KG has **6.2** improvements on macro F1 score, **5.1** improvements on micro F1 score, and **2.6** improvements on *p*@5 score against the strong baseline DR-CAML. We argue that this is because our proposed multi-scale attention can effectively capture better text representation and graph propagation can capture the coocuurence between codes. It is noticed that DR-CAML does not show many improvements over Text-CNN even though DR-CAML utilizes code descriptions as regularization and label-dependent attention. This reveals that large window size may reduce the effective information of local phrase, which causes the label-dependent attention not working. Our proposed multi-scale attention utilizes adaptive scale feature for phrase representation, resulting in a large improvement against fixed large window size.

4.5.2 Interpretability. In medical domain, it is important that model can provide explaination. We now evaluate the interpretability generated by our two-level attention mechanism against other baselines. In order to generate explaination, we select the most informative *n*-grams in the prediction of each label. We now describe how to extract informative *n*-gram phrase in detail: (1) Our proposed attention consists of multi-scale attention and label-depencent attention. We firstly use label-dependent attention mechanism to select the position i which is the argmax of the softmax output $\boldsymbol{\alpha}_l$ for prediction of l-th label, and then we use multi-scale attention to select scale size for position i which is the argmax of the softmax output $\boldsymbol{\alpha}_i$; (2) For CAML, we first use label-dependent attention to select the position same as our proposed method, and then we use receptive field which is window size to select most phrase for explaination; (3) For Text-CNN, we select the position that provides the maximum value of each channle at least once and weight by the final layer weights. More specifically, given the argmax vector $\boldsymbol{a}$ which is from max-pooling step as follows:

$$\begin{aligned} a_c &= \underset{j \in \{1,2,\cdots,m\}}{\text{argmax}} \ (\boldsymbol{x}_{cj}) \\ \boldsymbol{a} &= [a_1, a_2, \cdots, a_{d_c}] \end{aligned} \tag{11}$$

we can compute the importance of position i for l-th label as follows:

$$\alpha_{il} = \sum_{j:a_j=i} w_{l,j}^{cls} \tag{12}$$

where $w_{l,j}^{cls}$ is j-th scalar of final l-th label classifier weights. We then select most discriminative *n*-gram features for l-th label as $argmax_i(\alpha_{il})$.

We show the interpretability evaluation results in Table 6. It is observed that there are style gaps between official code description and highlighted phrase (e.g., *esophageal reflux* vs *feeding gastrostomy*). The code description and highlighted phrase describe same semantics but with different language style which causes EHR coding more difficult than traditional text classification. For the first example, only MSATT-KG successfully finds the informative phrase since "ETOH, depression" is related to respiratory failure. It reveals that MSATT-KG can effectively learn better representation benefitting from downstream feature reuse and composition. For the second example, both MSATT-KG and CAML find the discriminative sentences, however, our proposed MSATT-KG can accurately locate phrase "tracheostomy & feeding gastrostomy" while CAML only locates the long phrase which will cause information effusion

Table 6: Interpretability evaluation results for different models.

518.81:"Acute respiratory failure"	
MSATT-KG	...discharge diagnosis: left **hemothorax, ETOH, depression**, stable discharge condition...
CAML	...CXR showed persistent **small apical pneumothorax that remained unchanged, he is now tolerating a** regular diet...
Text-CNN	...serial chest x-rays **revealed a persistent left pleural effusion and due** to concern for loculated hemothorax...
530.81:"Esophageal reflux"	
MSATT-KG	...multiple rib fx requiring **tracheostomy & feeding gastrostomy**, fractures, acute renal failure, hypertension, GERD, anxiety cataracts, discharge condition mental status...
CAML	...right hemopneumothorax, multiple **rib fx requiring tracheostomy & feeding gastrostomy, fractures, acute renal failure, hypertension,** GERD, anxiety, cataracts...
Text-CNN	...major surgical or invasive procedure: **right thoracotomy, decortication of lung, mobilization of liver** off of chest wall...
37.23:"Combined right and left heart cardiac catheterization"	
MSATT-KG	...his dilated cardiomyopathy was secondary to tachycardia and **underwent cardiac catheterization** to evaluate for coronary disease....
CAML	...he was noted **to be in congestive heart failure. Lisinopril and digoxin were started**...
Text-CNN	...acute exacerbation of **systolic heart failure, dilated cardiomyopathy, severe mitral regurgitation**...

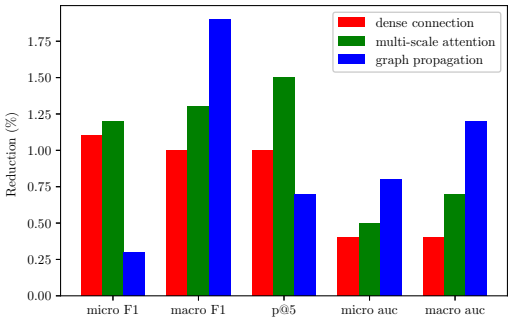
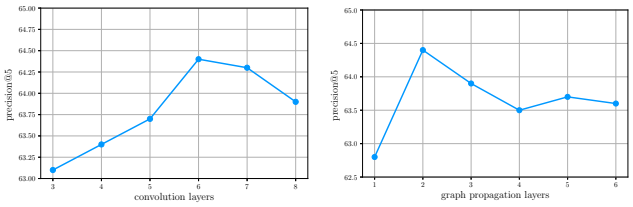


Figure 4: The effect of removing three components. We report the reduction amount on micro F1, macro F1, *p@5*, micro AUC, and macro AUC.



(a) Convolution layers **(b) Graph propagation layers**

Figure 5: Parameter sensitivity of MSATT-KG.

sometimes. Our proposed multi-scale feature attention can adaptively select most informative *n*-gram features rather than large window size which will introduce irrelevant information. For the prediction of code **37.23** which has fewer samples than the other two codes (**530.81**, **518.81**) at the last sample, only our proposed MSATT-KG successfully locates the correct phrase "underwent cardiac catheterization", while both CAML and Text-CNN locate the more common phrase "heart failure". It reveals that MSATT-KG can better adapt to the labels with only a few samples since proposed graph propagation helps learn better code classifiers which guide the attention mechanism via backpropagation.

4.6 Ablation Study

We perform ablation study to verify the effectiveness of each module in our model. That is, we remove each module from the model with ordinary replacement while without changing other modules. More specifically, the components we study here are dense connection, multi-scale attention, and structured knowledge graph propgation. For dense connection, we use stacked convolution with the same layers as an alternative; For multi-scale attention, we use the last convolutional layer's output as an alternative; For strcutured knowledge graph, we just use code description embeddings without graph propagation. We show the reduction amount of micro F1, macro F1, *p*@5, micro AUC, and macro AUC by removing each of three components in Fig 4. Firstly, it is observed that removing each component results in the reduce of all metrics, showing the effectiveness of these three components. Secondly, comparing the components with each other shows that multi-scale attention has the highest effect on micro F1 and *p*@5 while graph propagation has the highest effect on macro F1, micro AUC, and macro AUC. It reveals that the use of structured knowledge graph can capture the hierarchical relationships among medical codes and transfer knowledge among codes, which helps these codes with only a few samples to get better classifiers. Removing knowledge graph propagation decreases recall from 69.1% to 62.8%, although precision increases from 59.1% to 61.0%. On the other hand, our proposed multi-scale attention has the highest impact on these metrics which

treat each text code pair as a separate prediction since removing multi-scale attention decreases precision from 63.7% to 62.1%. Our proposed multi-scale attention and knowledge graph propagation are complementary thus our full model achieves 5.1 improvement on micro F1 score and 6.2 macro F1 score against CAML [12].

4.7 Parameter Sensitivity

In this section, we investigate the influence of parameters K and D in MSATT-KG. We vary K from 3 to 8 and D from 1 to 6, respectively, while keeping other parameters fixed. The results of $p@5$ on MIMIC-III dataset are presented in Fig 5. It can be observed from Fig 5a that, with the increase of K, the performance is boosted at first since more layers means more scale features, but drops after $K = 6$ because oversized n-gram features may introduce confusion which is harmful to our proposed multi-scale feature attention. Our proposed MSATT-KG achieves the best performance when $D = 2$ from Fig 5b. This is because graph propagation with a shallow layers cannot capture effective relationships, while a deep layers will result in over-smoothing between label embeddings.

5 CONCLUSION AND DISCUSSION

In this paper, we aim to predict diagnosis or procedure codes for clinical notes which we call EHR coding. We leverage a densely connected convolutional neural network to produce variable n-gram features. We also incorporate a multi-scale feature attention to adaptively select multi-scale features since the most informative n-gram in clinical notes for each word can vary in length according to the neighborhood. Furthermore, we leverage graph convolutional neural network to capture both the hierarchical relationship among medical codes and the semantics of each code to overcom the problem of extremely imbalanced label distribution. Our proposed method outperforms all state-of-the-art methods on both top-50 label setting and full-label setting, demonstrating the effectivness of our porposed modules.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China Projects No.U1636207, No.91546105, the Shanghai Science and Technology Development Fund No.16JC1400801, NSF under grants III-1526499, III-1763325, III-1909323, SaTC-1930941, and CNS-1626432.

REFERENCES

[1] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. 2018. Multi-label classification of patient notes: case study on ICD code assignment. In *Workshops at AAAI Conference on Artificial Intelligence*.
[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3 (2003), 1137–1155.
[3] Luciano R. S. de Lima, Alberto H. F. Laender, and Berthier A. Ribeiro-Neto. 1998. A Hierarchical Approach to the Automatic Categorization of Medical Documents. In *Proceedings of International Conference on Information and Knowledge Management*. 132–139.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[5] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.
[6] Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine* 65, 2 (2015), 155–166.
[7] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 1746–1751.
[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[9] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[10] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A Novel Bandit-based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* 18, 1 (Jan. 2017), 6765–6816.
[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[12] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*. 1101–1111.
[13] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2013. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association* 21, 2 (2013), 231–237.
[14] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2017. Condensed memory networks for clinical diagnostic inferencing. In *Proceedings of AAAI Conference on Artificial Intelligence*.
[15] Anthony Rios and Ramakanth Kavuluru. 2018. EMR Coding with Semi-Parametric Multi-Head Matching Networks. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, Vol. 1. 2081–2091.
[16] Elyne Scheurwegs, Boris Cule, Kim Luyckx, Léon Luyten, and Walter Daelemans. 2017. Selecting relevant features from the electronic health record for clinical code prediction. *Journal of biomedical informatics* 74 (2017), 92–103.
[17] Elyne Scheurwegs, Kim Luyckx, Léon Luyten, Walter Daelemans, and Tim Van den Bulcke. 2015. Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association* 23, e1 (2015), e11–e19.
[18] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards Automated ICD Coding Using Deep Learning. *arXiv preprint arXiv:1711.04075* (2017).
[19] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4077–4087.
[20] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of Association for Computational Linguistics and Joint Conference on Natural Language Processing*. 1556–1566.
[21] Ankit Vani, Yacine Jernite, and David Sontag. 2017. Grounded recurrent neural networks. *arXiv preprint arXiv:1705.08557* (2017).
[22] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *Proceedings of Association for Computational Linguistics*. 2321–2331.
[23] Sen Wang, Xiaojun Chang, Xue Li, Guodong Long, Lina Yao, and Quan Sheng. 2016. Diagnosis code assignment using sparsity-based disease correlation embedding. *IEEE Transactions on Knowledge & Data Engineering* 1 (2016), 1–1.
[24] Pengtao Xie and Eric Xing. 2018. A neural architecture for automated icd coding. In *Proceedings of Association for Computational Linguistics*, Vol. 1. 1066–1076.
[25] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Pengtao Xie, and Eric Xing. 2018. Multimodal Machine Learning for Automated ICD Coding. *arXiv preprint arXiv:1810.13348* (2018).
[26] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*. 1480–1489.
[27] Danchen Zhang, Daqing He, Sanqiang Zhao, and Lei Li. 2017. Enhancing Automatic ICD-9-CM Code Assignment for Medical Texts with PubMed. *BioNLP 2017* (2017), 263–271.