



HieNet: Bidirectional Hierarchy Framework for Automated ICD Coding

Shi Wang¹, Daniel Tang^{2(✉)}, Luchen Zhang³, Huilin Li⁴, and Ding Han⁵

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

wangshi@ict.ac.cn

² University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust (SNT), TruX, Luxembourg City, Luxembourg

xunzhu.tang@uni.lu

³ National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China

zlc@cert.org.cn

⁴ Department of Civil Engineering, Technical University of Denmark, 2800 Lyngby, Denmark

⁵ Huazhong University of Science and Technology, Wuhan, China

Abstract. International Classification of Diseases (ICD) is a set of classification codes for medical records. Automated ICD coding, which assigns unique International Classification of Diseases codes with each medical record, is widely used recently for its efficiency and error-prone avoidance. However, there are challenges that remain such as heterogeneity, label unbalance, and complex relationships between ICD codes. In this work, we proposed a novel Bidirectional Hierarchy Framework(HieNet) to address the challenges. Specifically, a personalized PageRank routine is developed to capture the co-relation of codes, a bidirectional hierarchy passage encoder to capture the codes' hierarchical representations, and a progressive predicting method is then proposed to narrow down the semantic searching space of prediction. We validate our method on two widely used datasets. Experimental results on two authoritative public datasets demonstrate that our proposed method boosts the state-of-the-art performance by a large margin.

Keywords: Structural encoder · ICD coding · Bidirectional passage retriever · Hierarchical embedding · Healthcare · Co-occurrence encoder

1 Introduction

The International Classification of Diseases (ICD) is widely considered as a healthcare multi-label classification system, supported by the World Health Organization (WHO). ICD codes have widely been used for reimbursement, taxonomy of diagnoses and procedures, and monitoring health issues [3, 20]. ICD coding needs coder to assign proper codes to a patient visit, which is composed of multiple long and heterogeneous textual narratives (e.g., discharge diagnosis, procedure notes, event notes), authored by different healthcare professionals, which means it's time-assuming, error-prone, and expensive in manual way. As a result, automated ICD encoding has attracted much attention

S. Wang and D. Tang—Represents equal contribution to this work.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13246, pp. 523–539, 2022.

https://doi.org/10.1007/978-3-031-00126-0_38

since it can save time and labor for billing. A number of neural network methods handling automated coding as multi-label task, have been proposed by [5, 19, 33], which converts the ICD coding into a set of binary classification for each code.

ICD codes can be organized in a tree-like structure. If a node N represents a kind of disease, the children of N are the sub-types of this disease. And in many cases, the differences among the siblings from one parent disease are very subtle. There are several challenges to link the discharge summaries with ICD codes: (1) Each admission record has a long and complex discharge summary. And the summaries are usually more than one thousand words long, containing medical history, diagnosis texts, surgical procedures, etc. Thus, it is difficult to assign proper codes to a given clinical note. (2) The label space to predict of ICD codes is very large (e.g., over 18,000 for ICD-9-CM) and the label distribution is extremely unbalanced. However, the average length of training-label codes in MIMIC-III-full is only 15.89 and the length of codes in 80% clinical notes is less than 22. Therefore, we only need to ensure the accuracy of top predicted codes. (3) Mutual exclusivity (*ME*) in ICD codes: As shown in Fig. 1), we use deep and light orange colors to represent *ME* codes. Deeper orange color represents ‘parent-child’ codes, and lighter orange color represents sibling codes. For example, given a clinical text, if a finer code ‘521.00’ is predicted, the parent of it ‘521.0’ and grand-parent ‘521’ should not occur in the predicted results. Furthermore, some sibling ICD codes should not appear in the same predicted result, such as ‘464.00’ (Acute laryngitis without mention of obstruction) and ‘464.01’ (Acute laryngitis with obstruction), because they are anti-sense. (4) Co-occurrence (*CC*) in ICD codes: We leverage light green and deep green colors for *CC* codes. Light green color represents reasoning co-occurrence codes. For example, ‘997.91’ (hypertension) usually leads to the occurrence of ‘429.9’ (heart disease, unspecified); Deep green color represents *CC* codes caused by common pre-conditions. For example, ‘staying up too late’ usually causes ‘997.91’ (hypertension) and ‘784.0’ (headache).

Some methods based on CNN [17, 19, 33] were proposed to address the issues from the characteristic (1) above, and they were proved efficient in extracting features from long texts. For characteristic (2), to our knowledge, no previous scheme was proposed to solve it. To address problem with characteristic (3), [32] leverage a sequential tree-lstm architecture to extract structural features of code tree. However, as we know, there is no contextual relationships among siblings in the code tree and it’s hard for tree-lstm process long sequential issue. Another approach [5] leverages hyperbolic ball to encode the *ME* feature, but it’s hard to measure the real performance of this non-euclidean method on *ME* problem. To address problem with characteristic (4), previous methods (e.g., [5]) employ GCN [14] to encode co-occurrence features of codes. However, GCN is unsuitable for describing the root node’s neighborhood and not designed the causal co-occurrence cases. In summary, issues from characteristics (2), (3), (4) remain to be solved.

In this paper, to address issues above, we present a novel method HieNet, which is short for Bidirectional Hierarchy Framework for Automated ICD Coding. HieNet contains three main modules, including a progressive mechanism (*PM* module), a bidirectional hierarchy passage encoder (*BHPE* module), and a personalized PageRank (*PP* module). These modules are designed as the solution of (2), (3), and (4), respectively.

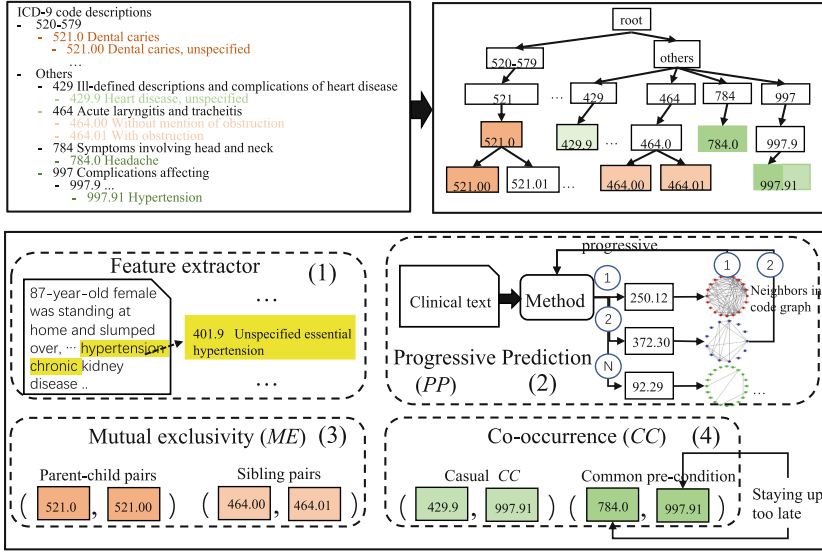


Fig. 1. A schematic diagram of the ICD code tree, matching process between clinical notes and codes (1), progressive mechanism (2), and relationships among ICD codes ((3) and (4)).

Progressive Prediction for (2): Given a clinical text or diagnose description, the previous methods predict all its ICD codes at the same time. However, some codes achieving high scores in their binary predictions can help predict other codes. For example, the neighbors of predicted code ‘401.9’ (unspecified essential hypertension) in code tree usually contains some other potential gold labels, such as ‘348.4’ (compression of brain). To make full use of first predicted, we introduce an approach named progressive mechanism to reduce the difficulty of improving the accuracy of predicted codes (e.g., average 14.3% improvement on Jaccard metric over the average scores of the best baselines).

Hierarchy Features Encoding for (3): There are two patterns in mutual exclusivity (*ME*): parent-child and sibling relationships. As stated above, ICD codes with *ME* relationships should not occur in one clinical note. To address the issue, we propose a bidirectional hierarchy passage encoder (*BHPE*) that contains two sub-modules: bidirectional passage retriever (*BPR*) and tree position encoder (*TPE*). The experimental results indicate that the *BHPE* module improves HieNet by 12.0% on macro-F1 on MIMIC-III full.

Code Co-Occurrence Encoding for (4): Some codes have causal or pre-condition co-occurrence relationship, which is called code co-occurrence (*CC*). Pre-condition *CC* codes are usually caused by common bad habits or hurts. We propose personalized PageRank (*PP*) to encode pre-condition *CC* features. Furthermore, the combination of *PM* and *PP* enable *PP* has the ability to encode causal *CC* relationships among ICD

codes. The experimental results show that *PP* module makes the 10.7% improvement on macro-F1 on MIMIC-III full and 13.1% improvement on macro-F1 on MIMIC-II.

Our Contributions: 1) To the best of our knowledge, we are first to propose the progressive mechanism to improve the accuracy of top_K predicted ICD codes. 2) We are first to introduce the bidirectional passage retriever and tree position encoder as the solutions of two patterns (i.e., parent-child, sibling) of mutual exclusivity. 3) We introduce a personalized PageRank to encode the pre-condition *CC* and leverage progressive mechanism to capture casual *CC* features. 4) The experimental results on two widely used datasets illustrate that HieNet outperforms the state-of-the-art compared to the previous methods (e.g., 20.6% improvement on top_{30} -Jaccard (MIMIC-III 50) over the best baseline CAML).

2 Related Work

Automatic ICD Coding: Automatic ICD coding has been studied in a large coverage of areas, including information retrieval, machine learning, and healthcare. [12] treat the ICD coding as a multi-label text classification problem and introduced a label ranking approach based on the features extracted from the clinical notes. [19] proposed the landmark work CAML with attention algorithm and leveraged CNN to capture the key information for each code, and DR-CAML is a updated version of CAML with code description proposed in the same publication. Inspired by CAML, more CNN-based methods are proposed, including [2, 26, 36]. [28] explored character based on LSTM with attention and [32] applied tree LSTM with hierarchy information for ICD coding. [35] applied multi-modal machine learning to predict ICD codes. [31] transferred the ICD coding into a path generation task and developed an adversarial reinforcement path generation framework for ICD coding.

Hierarchical Encoder: Tree position encoder was proposed by [7] and applied in tree-based transformers. Recent works demonstrate that tree position encoder can handle source code summarization [1] and semantic prediction [8]. To our knowledge, we are the first to take the tree position encoder to capture the position embedding of code tree.

Co-occurrence Encoder: PageRank was proposed by [23] and widely employed in website page ranking [16], latent topics mining [22], key phrase extraction [7], and multilingual word sense disambiguation [27]. [15] proposed personalized PageRank (PPNP) to address the limit distribution of GCN. We are the first to use the personalized PageRank to encode the code co-occurrence representations for the automated ICD coding task.

3 Proposed Model

3.1 Problem Definition

Following the previous works [5, 32], We formulate ICD coding as a multi-label text prediction problem. We make some definitions here to help state the proposed work well.

- **Definition 1: CC.** CC is short for Code co-occurrence and it contains three cases: (1) pre-condition CC. Intuitively, some diseases are usually caused by common bad habits; (2) causal CC. (3) other co-occurrence codes without obvious reasons.
- **Definition 2: ME.** ME is short for Mutual exclusivity and it contains two cases: (1) Parent-child pairs. (2) Complementary sibling pairs, such as ‘464.00’ (without mention of obstruction) and ‘464.01’ (with obstruction).
- **Definition 3: DP.** DP represents improving the accuracy of top_K predicted codes. Actually, 80% clinical notes have less than 22 codes, so we need to focus on improving the accuracy of top_K predicted codes.

3.2 Model Architecture

Figure 2 shows an overview of the bidirectional hierarchy passage framework. Firstly, we encode the code hierarchy semantic as hierarchical code representations via bidirectional hierarchy passage retriever. Furthermore, we employ a multi-channel CNN to obtain clinical document embeddings and conduct a code-wise operation between document embeddings and hierarchical code representations. Secondly, we introduce a progressive mechanism to improve the accuracy of top_K predicted ICD codes (DP). Thirdly, we leverage a personalized PageRank algorithm to calculate the co-occurrence relationships among ICD codes. Finally, we aggregate the results of above modules and conduct a full connected layer with multiple sigmoid functions to generate 0–1 probability distributions for each code.

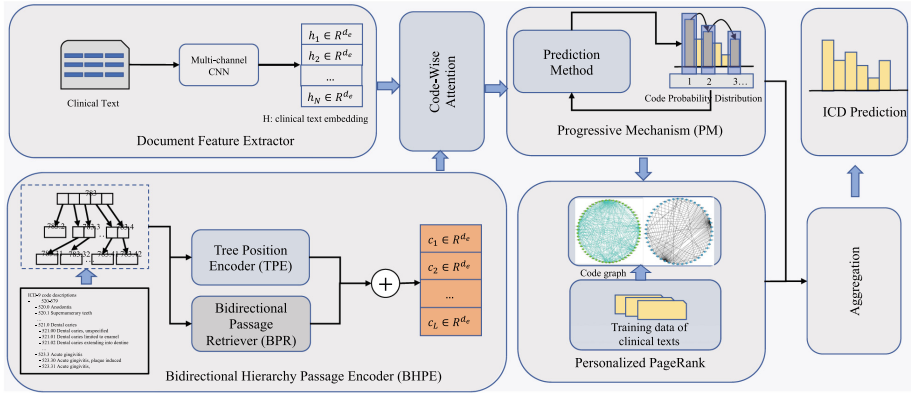


Fig. 2. The architecture of our model.

3.3 Document Feature Extractor (DFE)

Given a electronic health record $\mathbf{W} = \{w_1, w_2, \dots, w_N\}$ (N donates the length of \mathbf{W}), we map \mathbf{W} into a vector representation $\mathbf{X} \in \mathbb{R}^{d_e \times N}$ where d_e indicates the dimension of word embedding. We leverage a multi-channel one-dimensional convolution neural network to encode clinical texts. A convolution operation contains a filter \mathbf{W}_f with a

window of k words. For example, a feature f_i is generated with a window of words $x_{i:i+k-1}$ by

$$f_i = \text{relu}(W_f * x_{n:n+k-1} + b_c) \quad (1)$$

where $\text{relu}(\cdot)$ is a non-linear transformation function, $b \in \mathbb{R}^{d_e}$ is a bias, $*$ is the concatenation operator. This filter is applied with multiple filter size to produce a final feature map f :

$$f = f_{1,2,\dots,n-l+1} = f_1 \oplus f_2 \oplus \dots \oplus f_{n-l+1}. \quad (2)$$

Then we employ a pooling operation over feature map f and take the maximum value as the final value of f by $\hat{f} = \max(f)$. Regarding l channels (l different window sizes), we concatenate generated l feature maps as a representation H of an clinical text as follows:

$$H = \hat{f}_1 \oplus \hat{f}_2 \oplus \dots \oplus \hat{f}_{n-l+1}. \quad (3)$$

3.4 Bidirectional Hierarchy Passage Encoder (BHPE)

This section will introduce the hierarchy encoder that includes two main modules: bidirectional passage retriever (**BPR**) and tree position encoder (**TPE**). First, we construct a dimensional vector by averaging the vectors of words of code's description to represent the code. Then, we propose **BPR** to capture parent-child relationships. Moreover, we introduce **TPE** to encode tree positions of codes. Next, we add representations from **BPR** and **TPE** as the final vectors of codes that contain both hierarchical and parent-child contextual features. Finally, we obtain code-wise document representations by conducting code-wise attention between codes vectors and document representations.

Bidirectional Passage Retriever (BPR). BPR uses two independent BERT encoders to capture the hierarchical relationships among ICD codes, including two directional process: up-stream u (child \rightarrow parent) and down-stream d (parent \rightarrow child):

$$e_u = \text{BERT}_u(u), e_d = \text{BERT}_d(d), \quad (4)$$

where $e_u \in \mathbb{R}^d$ and $e_d \in \mathbb{R}^d$. We use the uncased version of BERT-base; therefore, $d = 768$. The initial embedding tool of BERT we employ is WordPiece tokenization (wp) which is different from the initial method of ICD codes (word2vec)¹.

In the up-stream passage retriever, an internal node (with M children) is comprised of these components: a position cell p for each node, a self-input cell i_\uparrow , and a BERT cell $\{b_\uparrow\}_{m=1}^M$ for M children. The position cell is used to encode the related relationship of ICD codes in the hierarchical code tree and the computation of p is shown in Sect. 3.4. The transition equations of among components are:

$$\begin{aligned} u &= \text{Set}(p_k + wp(w_k))_{m=1}^M, \\ i_\uparrow &= \{b_\uparrow\}_{m=1}^M = \text{BERT}_u(u), \end{aligned} \quad (5)$$

where w_k is the k -th token of M children of node C .

¹ Why initial methods of ICD codes and clinical codes are diffident? Answer: ICD codes are usually beyond the vocabulary of BERT because they are professional and technical terms while words in clinical are original that can be covered by the vocabulary of BERT. Therefore, for better representing ICD codes and clinical notes, we leverage word2vec tool and WordPiece tokenization function in BERT to init them, respectively.

In the down-stream passage retriever, for a not-root node, it has such one component: an input cell i_{\downarrow} . The transition equation is:

$$i_{\downarrow} = BERT_d(p + wp(w)) \quad (6)$$

where p is the position embedding of the parent node and w is the average embedding of the node's description. Since root node has no parent, i_{\downarrow} cannot be computed using the above equations. Instead, we set i_{\uparrow} to i_{\downarrow} here.

Loss function of BPR: We call a pair of one up-stream *PR* and one down-stream *PR* an interaction. In one interaction, one node n_j has two representations: $n_j^{i_{\uparrow}}$ and $n_j^{i_{\downarrow}}$. The goal of training **BPR** is to reduce the difference between $n_j^{i_{\uparrow}}$ and $n_j^{i_{\downarrow}}$ as much as possible. Both $n_j^{i_{\uparrow}} (X = \{0, 1, 2, \dots, k, \dots, d_e\})$ and $n_j^{i_{\downarrow}} (X)$ can be recognized as two distributions. So we construct a Kullback-Leibler divergence [25] as the loss function of **BPR**:

$$\begin{aligned} \mathcal{L}_{bpr1} &= \frac{1}{L} \sum_1^L \sum_1^{d_e} [n_j^{i_{\uparrow}}(x) \log(n_j^{i_{\uparrow}}(x)) - n_j^{i_{\uparrow}}(x) \log(n_j^{i_{\downarrow}}(x))] \\ \mathcal{L}_{bpr2} &= \frac{1}{L} \sum_1^L \sum_1^{d_e} [n_j^{i_{\downarrow}}(x) \log(n_j^{i_{\downarrow}}(x)) - n_j^{i_{\downarrow}}(x) \log(n_j^{i_{\uparrow}}(x))] \\ \mathcal{L}_{bpr} &= \mathcal{L}_{bpr1} + \mathcal{L}_{bpr2} + \|\mathcal{L}_{bpr1} - \mathcal{L}_{bpr2}\|_2^2 \end{aligned} \quad (7)$$

where L is the number of all ICD codes, d_e represents the dimension of ICD codes, n_j is the j -th node of ICD codes, x represents x -th item of n_j , \mathcal{L}_{bpr1} represents child-to-parent KL value between $n_j^{i_{\uparrow}}$ and $n_j^{i_{\downarrow}}$, \mathcal{L}_{bpr2} represents parent-to-child KL value between $n_j^{i_{\uparrow}}$ and $n_j^{i_{\downarrow}}$.

When the loss value is minor than 0.01, the training stops. Finally, we obtain parent-child code representations $Vt \in \mathbb{R}^{d_e \times N}$.

Tree Position Encoder (TPE) . Inspired by [29], we propose a tree position encoder to capture the positional features of ICD codes.

$$PE_{\beta} = A_{\phi} PE_{\alpha} \quad (8)$$

Given a position α , we can get the target position β with affine transform operation A_{ϕ} (Eq. (8)). The path between α and β can be considered as the set of n length-1 paths. Directions of these paths contains "to parent" and "to children". We take "to parent" as function U and "to children" function D . Thus, for any path ϕ , we can obtain the transform A_{ϕ} by some combinations of U and D . For example, the position encoding of the second child of node x 's grandpa can described path $\phi = \langle \text{parent, parent, child-2} \rangle$, which can also be represented as $D_2 U^2 P E_x$.

We take the root node as zero vector ($\mathbf{0} \in \mathbb{R}^{d_e}$). Then, since all the paths from the root to other positions are downward, we can get embedding of any position x via the Eq. (9).

$$x = D_{b_L} D_{b_{L-1}} \dots D_{b_1} \quad (9)$$

where L here means the L -th layer the node is located, and b_i represents the chosen path in the i -th layer. We treat tree position encoding as a stack of length-1 component parts.

Every D operation pushes a length-1 path onto the stack, while U pops a length-1 path, which is described as Eq. (10):

$$\begin{aligned} D_i x &= e_i^n \oplus (x \ominus x[n+1 :]) \\ Ux &= (x \ominus x[: (n-1)]) \oplus \mathbf{0}_n \end{aligned} \quad (10)$$

where \ominus means pop operation or truncation, and \oplus indicates push or concatenate operation. In addition, e_i^n is an one-hot encoding with n elements and n is the total number of children of a parent. An example of function D is shown in Fig. 3.

After tree-position encoding, we obtain code tree-position embedding ($Vp \in \mathbb{R}^{d_e * N}$). Then add Vp to Vt to get final code representation V_{pt} that contains both parent-child relationships and hierarchical features.

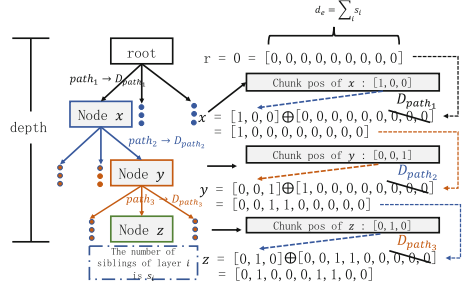


Fig. 3. An example of function D

Code-Wise Attention. Inspired by [5], we use code-wise attention to generate code-aware document representations by using representations outputted from document feature extractor and bidirectional hierarchy passage encoder. The code-wise attention feature a_l for code l is calculated by:

$$\begin{aligned} s_l &= \text{Softmax}(\tanh(H \cdot W_a^T + b_a) \cdot v_{ptl}) \\ a_l &= s_l^T \cdot H \end{aligned} \quad (11)$$

where Softmax is the normalized exponential function, s_l donates the attention scores for all elements in document representation H , a_l represents the most relevant information in H about the code l by code-wise attention. Then we get $d_e \times L$ dimensional code-wise adjacent document representations.

3.5 Progressive Mechanism (PM) for DP

This section introduces a simple progressive mechanism to address the **DP** problem in ICD coding. A former predicted code could help to predict next codes. For example, the neighbours of ‘Diabetes mellitus’ contain ‘heart disease’, neighbours of ‘heart disease’ contain ‘cardiovascular disease’, then there exists a prediction path <Diabetes mellitus, heart disease, cardiovascular disease>. Given a clinical note, if ‘Diabetes mellitus’ is true, then we could use this label to predict ‘heart disease’. Similarly, we can use ‘heart disease’ to predict ‘cardiovascular disease’. But if ‘heart disease’ is wrong, the process of progressive prediction will end. The problem is how to use the prior predicted code to predict other codes. We address the problem above by constructing an average-operation between the prior code’s hidden output and other codes’ hidden outputs with

a hyper-parameter λ . Assuming the hidden outputs are described as $f_i \in \mathbb{R}^{1 \times 1}$ and it is proved true, then later hidden output f_j should be calculated as follows:

$$f_j = \lambda f_i + (1 - \lambda) * f_j \quad (12)$$

where λ is a trade-off factor to balance f_j and f_i . Here, f_i decides the influence from a former code to a later code. f_j reflects the current value of node j itself.

Figure 4 shows the dynamic progressive prediction in detail. After PM , we obtain the output hidden embedding of ICD coding methods $P = \{p_1, p_2, \dots, p_L\} \in \mathbb{R}^{d_e * L}$.

3.6 Personalized PageRank (PP) for CC

Inspired by $PPNP$ [15], we build a personalized PageRank [23] to capture the code co-occurrence. Given a clinical text and its golden labels, we can build a strong connected sub-graph, and connect all sub-graphs into a big graph $G = (V, E)$, where V and E are sets of nodes (codes) and edges respectively. Let L denote the number of nodes and m the number of edges. The graph G is described by the matrix $A \in \mathbb{R}^{L * L}$, and $\tilde{A} = A + I_n$ denotes the matrix A with added self-loops.

In the graph G , let the initial representation of node x as i_x . In addition, We can update node embedding by the recursion equation:

$$\begin{aligned} PPR(i_x) &= (1 - d)\tilde{A} + di_x \\ &= d(I_n - (1 - d)\tilde{A})^{-1}i_x \end{aligned} \quad (13)$$

where $\tilde{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$. D is a degree matrix of nodes and \tilde{D} denotes the degree matrix of graph with added self-loops, where $\tilde{D}_{i,i} = \sum_j \tilde{A}_{i,j}$.

We generate predictions based on each node's own characteristics, and then propagate them through a fully personalized PageRank scheme to generate the pagerank-aware prediction. The pagerank-aware embedding is denoted as $PPR = \{ppr_1, ppr_2, \dots, ppr_L\} \in \mathbb{R}^{d_e * L}$.

3.7 Aggregation and Training

Aggregation: After exploiting the structural information of code via **BHPE** (i.e., **BPR** and **TPE**) and code co-occurrence via personalized PageRank, we obtain a normal representation (P) (from PM in Sect. 3.5) and a pagerank-aware representation (PPR) (Sect. 3.6), respectively. We concatenate P and PPR as $P3R \in \mathbb{R}^{d_e * 2L}$ and conduct a fully connected layer to reshape the matrix in to $\mathbb{R}^{d_e * L}$ embedding space.

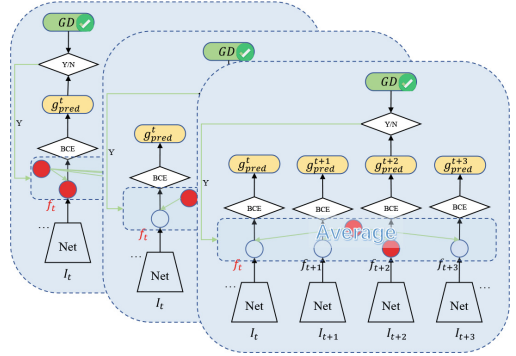


Fig. 4. Overview of progressive mechanism, *Net* represents the network of method for automated ICD coding, *BCE* is binary cross-entropy, *GD* is short for Ground Truth.

Training: Since the automated icd coding is a multi-label prediction task, we employ a multi-label binary cross-entropy (*BCE*) as the loss of our model:

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\sum_{l=1}^L [y_l \log(\hat{y}_l) + (1 - y_l) \log(1 - \hat{y}_l)] \quad (14)$$

where y_l is the ground truth and \hat{y}_l is the predicted value, $\hat{y}_l = \sigma(x_l)$. Here, we use Adam optimizer [13] to propagate the parameters of our model.

4 Experimental Setup

4.1 Datasets

MIMIC-II [11] and MIMIC-III [10] are the most widely open-access datasets for evaluating automated ICD encoding methods. In MIMIC-III, there exists two versions. One is MIMIC-III full and the other is MIMIC-III 50. For MIMIC-III full, there are 8,921 unique codes, 47,723 discharge summaries for training, 3,372 summaries for test, and 1,631 for validation. For MIMIC-III 50, we use a set of 8,066 for training, with 1,729 summaries and 1,729 summaries for validation and test, respectively. In MIMIC-II dataset, there are 5,031 clinical codes, and we use the same experimental setting as previous works [4, 5, 24].

4.2 Metrics and Parameter Settings

We use macro/micro-averaged F1, macro/micro-averaged AUC, and P@N as the main metrics to evaluate our model and baselines. To measure **DP**, we use Jaccard Similarity Coefficient [21] as the metric, which is defined as $Jaccard = \frac{1}{m} \sum_i^m |Y_i \cap \hat{Y}_i| / |Y_i \cup \hat{Y}_i|$, where m is the number of instances of the dataset, Y_i is the predicted results from different ICD coding methods, \hat{Y}_i indicates the ground truth ICD set (**note:** Assuming that $\text{len}(Y) = l_y$, $\text{len}(\hat{Y}) = K$, $\text{len}(Y \cap \hat{Y}) = l_{\cap}$, if $K \geq l_y$, let $\hat{Y} = \hat{Y}[:m]$, else let $Y = (Y \cap \hat{Y}) \cup \mathbf{0}^*(K - l_{\cap})$).

We set the word embedding size d_e as 100. The size of hidden layer is 128. We set 5 channels in CNN and the filter-sizes of them are 1, 3, 5, 7, 10, respectively. The dropout rate is 0.2. The learning rate is $1e^{-4}$. The batch size is set as 32. The dump rate d in Eq. (13) is 5. The maximize of personalized PageRank loop is 50. The criterion for early stopping is 10. The initial embedding tool for clinical codes is word2vec [18]. The initial embedding tool for clinical text is WordPiece tokenizer in BERT. The reason of why we use different initial embedding methods for clinical codes and nodes are shown in footnote 1 in Sect. 3.3.

4.3 Baselines

In order to demonstrate the effectiveness of HieNet, we compare it with several previous methods, including state-of-the-art models with knowledge graph and GCN.

- **CNN.** CNN is a widely applied method in language modeling [6].

- **CAML & DRCAML** [19]. CAML leverages convolutional attention for automated ICD prediction. DR-CAML is a updated version of CAML with code description.
- **HyperCore** [5]. This method employed hyperbolic representation to capture the code hierarchy and used GCN [14] to encode the semantic of the code co-occurrence.
- **JointLATT** [30]. This method proposed a new label attention method. And the label attention model achieve SOTA results compared with other previous works.
- **MASTATT-KG** [34]. This method utilizes a multi-scale feature attention to select multi-scale features adaptively.
- **MultiResCNN** [17]. MultiResCNN contains a multi-filter convolutional layer to capture various text patterns and a residual convolutional layer to enlarge the receptive field.
- **DCAN** [9]. This method proposes a dilated convolutional attention network, integrating dilated convolutions, residual connections, and label attention, for medical code assignment.

5 Result and Analysis

We focus on answering the following **researching questions (RQs)**:

- **RQ1:** What is the performance of HieNet on ICD encoding task?
- **RQ2:** Why progressive mechanism and how it performs?
- **RQ3:** What are the contributions of the different components?
- **RQ4:** How does the trade-off coefficient (λ in Sect. 3.5) influence the performance?

5.1 Overall Performance (RQ1)

The comparisons between our model and other state-of-the-art on MIMIC-II and MIMIC-III are given in Tables 1 and 2, respectively. Our HieNet model outperforms every single baseline on most of metrics. The CAML architecture is comparable to the DR-CAML, and the CNN baseline essentially performs the worst than all other neural architectures. We recognized P@N as the most intuitive measure to indicate the effectiveness of methods, since it examines the ability of the method to return a high-confidence subset of codes. Moreover, Jaccard is used to measure the accuracy of top_K predicted codes.

For MIMIC-III Full: Compared with baselines, HieNet achieves the best performance on macro-F1, micro-F1, and macro-AUC. Since clinical codes are in uneven distribution and macro-F1 emphasizes the performance of rare label, it is difficult to obtain high macro-F1 score. Even in this case, HieNet performs perfectly and achieves 3% improvement compared to the latest state-of-the-art HyperCore method. This demonstrates the effectiveness of HieNet. Furthermore, on Jaccard metric, HieNet improves the performance by a big margin with 7.7% improvement on top_{20} (from 33.7% to 36.3%) and 19.0% improvement on top_{30} (from 23.1% to 27.5%).

For MIMIC-III 50: Following the previous work [5, 19], we also evaluate our model and baselines on the most common 50 codes set of MIMIC-III. Different from

Table 1. Results (%) of the comparison of our model and other baselines on the MIMIC-III full and MIMIC-III 50. In all tables, the bold number with * indicates the best result compared to other methods.

	MIMIC-III full							
Model	Jaccard		AUC		F1		P@N	
	top_20	top_30	Macro	Micro	Macro	Micro	8	15
CNN	30.2	20.9	80.6	96.9	4.2	41.9	40.2	49.1
CAML	32.4	22.5	88.8	98.4	7.1	51.9	69.7	54.9
DR-CAML	33.7	23.1	89.7	98.5	8.6	52.9	69.0	54.8
HyperCore	-	-	93.0	98.9	9.0	55.1	72.2	57.9
JointLAAT	-	-	92.1	98.8	8.9	55.3	73.5	59.0
MSATT-KG	32.1	22.0	91.0	99.2	9.0	55.3	72.8	58.1
MultiResCNN	-	-	91.0	98.6	8.5	55.2	73.4	58.4
HieNet	36.3* ± 0.2	27.5* ± 0.3	93.3* ± 0.4	99.2* ± 0.2	9.3* ± 0.1	56.6* ± 0.7	78.3* ± 0.5	65.0* ± 0.3
	MIMIC-III 50							
Model	Jaccard		AUC		F1		P@N	
	top_20	top_30	Macro	Micro	Macro	Micro	5	-
CNN	31.7	24.3	87.6	90.7	57.6	62.5	62.0	-
CAML	32.5	25.2	87.5	90.9	53.2	61.4	60.9	-
DR-CAML	32.5	24.6	88.4	91.6	57.6	63.3	61.8	-
HyperCore	-	-	89.5	92.9	60.9	66.3	63.2	-
JointLAAT	-	-	92.5	94.6	66.1	71.6	67.1	-
MSATT-KG	33.9	24.7	91.4	93.6	63.8	68.4	64.4	-
MultiResCNN	-	-	89.9	92.8	60.6	67.0	64.1	-
DCAN	-	-	90.2	93.1	61.5	67.1	64.2	-
HieNet	37.7* ± 0.1	30.4* ± 0.2	93.4* ± 0.8	95.0* ± 0.1	67.1* ± 0.2	72.4* ± 0.2	69.5* ± 0.3	-

MIMIC-III full, MIMIC-III 50 has a relatively smooth distribution, which leads to the possibility of achieving higher macro-F1 scores.

Our method obtains the highest score on macro-AUC, micro-AUC, and P@5 metrics. On MIMIC-III 50, HieNet achieves the best performance on most of the evaluation except micro-F1 and macro-F1. The top_{30} -Jaccard value gets the most significant improvement (i.e., 20.6% over the best baseline CAML). The reason is that codes on datasets on MIMIC-III 50 are more closely related, and progressive mechanism (CC) is just designed for this.

For MIMIC-II: As shown in Table 2, MIMIC-II contains 5,031 labels, and our method HieNet also performs the best on most metrics compared with baselines except macro-F1 value. In addition, latest work HyperCore’s macro-AUC, micro-AUC, micro-F1, and P@8 are much lower than HieNet (1.0%, 1.2%, 3.1%, and 5.4% lower). top_{20} and top_{30} Jaccard values indicate that CNN, CAML, and DR-CAML are poor in leveraging relationships among codes while HieNet can predicts a higher coverage of correct ICD codes.

Table 2. Experimental results of our model and other baselines on MIMIC-II.

Model	Jaccard		AUC		F1		P@N
	top_{20}	top_{30}	Macro	Micro	Macro	Micro	
CNN	13.7	11.3	74.2	94.1	3.0	33.2	38.8
CAML	14.4	12.3	82.0	96.6	4.8	44.2	52.3
DR-CAML	13.8	11.7	82.6	96.6	4.9	45.7	51.1
HyperCore	-	-	88.5	97.1	7.0	47.7	53.7
JointLAAT	-	-	87.1	97.2	6.8	49.1	55.1
MultiResCNN	-	-	85.1	96.8	5.2	46.4	54.4
HieNet	15.7* \pm 0.1	13.2* \pm 0.3	89.4* \pm 0.2	98.3* \pm 0.1	7.1* \pm 0.5	49.2* \pm 0.2	56.6* \pm 0.3

5.2 Effectiveness of Progressive Mechanism (RQ2)

We take an example in Fig. 5. For a given clinical text (patient 118299 here), we verify the impact of the first predicted code on predicting the second one, and the effect of the front two predicted codes on predicting the third one.

The gold label of patient 118299 is $gl = ['198.3', '348.5', '162.9', '401.9', '272.4', '15.9', '22.0', '20.5']$, and the front predicted codes are '401.9' and '15.9'. The node '401.9' has 1,655 neighbors and Fig. 5a just shows the 20 of them. The node '15.9' has 56 neighbors and Fig. 5b only shows 20 of them. The shared neighbors (i/t -1&2 in Fig. 5c) are i/t -1&2 = ['041.19', '22.1', '433.31', '209.79', '013.25', '237.6', '239.6', '43.11', '372.30', '33.22', '23.4', '803.62', '20.5']. As observed, '20.5' occurred in i/t -1&2 is exactly one element in gl . Thus, we only need 13 labels (i/t -1&2) to predict the third code.

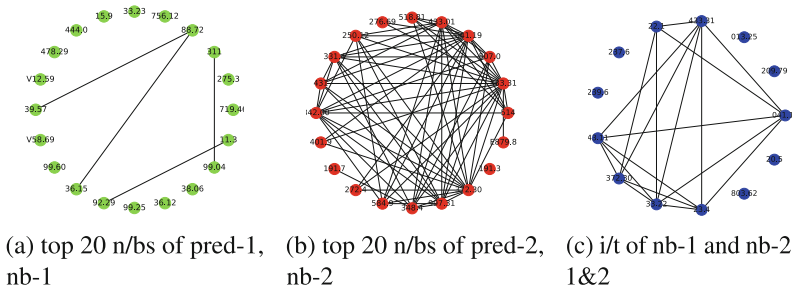


Fig. 5. Example of progressive mechanism in HieNet on patient 118299 summaries. n/bs means neighbors, i/t represents interaction, pred-1 indicates the top 1 of predicted codes, nb- i denotes the set of neighbors of i -th predicted codes, and it -1&2 represents the interacted result of nb-1 and nb-2.

Obviously, the neighbors of the former predicted codes can be used help predict the one. In conclusion, the process of above demonstrates the effectiveness of progressive mechanism.

5.3 Ablation Study (RQ3)

We conduct ablation investigation to examine the effectiveness of each part in our model. To evaluate a model, we remove it (denoted as without, w/o) and perform the remaining part on the datasets. The experimental results of ablation study are shown in Table 3.

Table 3. Ablation study by removing the main components.

Models	MIMIC-III full		MIMIC-III 50		MIMIC-II	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
HieNet	9.3*	56.6*	67.1*	72.4*	6.9*	49.2*
w/o <i>PM</i>	8.9	56.3	59.8	64.1	6.5	47.7
w/o <i>BHPE</i>	8.3	55.2	57.7	62.6	6.1	47.5
w/o <i>PP</i>	8.4	56.1	58.6	65.3	6.5	47.3
w GCN for <i>CC</i> (w/o <i>PP</i>)	9.1	55.3	60.4	65.7	6.7	49.0
w tree-lstm for <i>DP</i> (w/o <i>BHPE</i>)	8.8	55.2	60.0	64.6	6.6	47.9

Impact of *PM*. We remove the *PM* part from the full model. As shown in Table 3, HieNet without progressive module achieves lower scores of macro-F1, micro-F1 on both MIMIC-II and MIMIC-III.

Impact of *PP*. Compared with the w/o personalized PageRank module, the full HieNet improves the score on Macro-F1 (MIMIC-III full) from 0.084 to 0.093, 0.586 to 0.611 (MIMIC-III 50), and 0.065 to 0.069 (MIMIC-II), respectively. As shown in Table 3, the line ‘w GCN for *CC* (w/o *PP*)’ performs poorer than HieNet.

Impact of *BHPE*. We remove *BHPE* and compare with the full HieNet. The result is given in Table 3, HieNet improves 10.7% (from 0.084 to 0.093) on macro-F1 (MIMIC-III full), and 14.5% (from 0.586 to 0.671) on macro-F1 (MIMIC-III 50), 6.2% (from 0.065 to 0.069) on macro-F1 (MIMIC-II), etc., respectively.

Table 3 shows that HieNet with tree-lstm instead of *BHPE* performs worse on all metrics on three main datasets. The results from Table 3 demonstrates the effectiveness of different modules in HieNet. In addition, *BHPE* module plays a more important role in HieNet compared with *PM* and *PP*.

5.4 The Impact of λ (RQ4)

The trade-off parameter λ is used to balance the influence factor from former hidden outputs and current hidden output. When λ is larger, HieNet relies more on the former predicted codes to influence the next ones to predict, which means there exists a strong reasoning relationship in the predicted codes corresponding to the given clinical note. When λ is smaller, HieNet tends to prioritize the current hidden output to predict a proper ICD code during the learning. The value of λ is set from (0, 0.1, 0.2, ..., 1.0) to measure the performance. The results measured on MMIMIC-III-full are shown in Fig. 6.

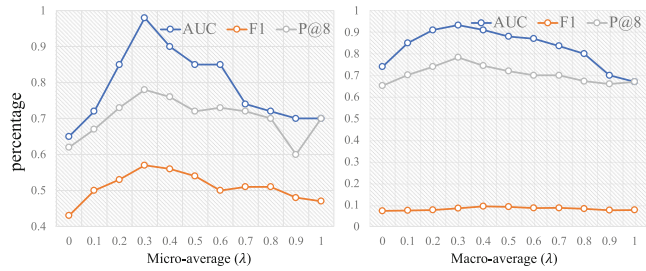


Fig. 6. Performance with different values of λ .

First, as λ increases, all metrics increase gradually at the beginning, but most of them decrease when $\lambda \geq 0.3$. In addition, F1 of macro-average decreases after $\lambda \geq 0.4$. The best performance is not achieved when $\lambda = 0$ or 1. This demonstrates that progressive mechanism improve the performance of ICD coding methods. Second, Fig. 6 shows that the beginning increases of macro-F1, macro-AUC, micro-F1, and micro-AUC are faster than the afterward decreases. Specially, micro-AUC grows from 0.65 to 0.98 only by the 0.3 added λ -value while micro-AUC drops from 0.98 to 0.7 needs another 0.7 added λ -value. This indicates the information of prior predicted codes can significantly affect the performance of ICD coding.

6 Conclusion and Future Works

In this paper, we propose the HieNet, which employs multi-channel CNN to encode the document representation, bidirectional hierarchy encoder to capture the hierarchy features, progressive mechanism improve the accuracy of top_K predicted codes, and personalized PageRank to obtain code co-occurrence. HieNet yields strong improvements over previous methods, while providing the new state-of-the-art performance on both MIMIC-II and MIMIC-III.

However, for input, the future works need to pay more attention to multi-type input. The input can be medical image (i.e., chest radio graph), structured information (i.e., prescriptions), unstructured data (i.e., clinical texts), etc. Inspired by the effectiveness of the progressive mechanism, we can build a multi-model prediction model in future.

References

1. Ahmad, W.U., Chakraborty, S., Ray, B., Chang, K.: A transformer-based approach for source code summarization. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) ACL 2020, pp. 4998–5007. Association for Computational Linguistics (2020)

2. Allamanis, M., Peng, H., Sutton, C.: A convolutional attention network for extreme summarization of source code. In: Balcan, M., Weinberger, K.Q. (eds.) ICML 2016. JMLR Workshop and Conference Proceedings, vol. 48, pp. 2091–2100. JMLR.org (2016). <http://proceedings.mlr.press/v48/allamanis16.html>
3. Avati, A., Jung, K., Harman, S., Downing, L., Ng, A.Y., Shah, N.H.: Improving palliative care with deep learning. BMC Med. Inform. Decis. Mak. **18**(S-4), 55–64 (2018)
4. Baumel, T., Nassour-Kassis, J., Elhadad, M., Elhadad, N.: Multi-label classification of patient notes a case study on ICD code assignment. CoRR abs/1709.09587 (2017)
5. Cao, P., Chen, Y., Liu, K., Zhao, J., Liu, S., Chong, W.: HyperCore: hyperbolic and co-graph representation for automatic ICD coding, pp. 3105–3114. ACL, Online, July 2020. <https://doi.org/10.18653/v1/2020.acl-main.282>, <https://www.aclweb.org/anthology/2020.acl-main.282>
6. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Precup, D., Teh, Y.W. (eds.) ICML 2017. Proceedings of Machine Learning Research, vol. 70, pp. 933–941. PMLR (2017)
7. Florescu, C., Caragea, C.: A position-biased PageRank algorithm for keyphrase extraction. In: Singh, S.P., Markovitch, S. (eds.) AAAI 2017, pp. 4923–4924. AAAI Press (2017)
8. Huber, P., Carenini, G.: From sentiment annotations to sentiment prediction through discourse augmentation. In: Scott, D., Bel, N., Zong, C. (eds.) COLING 2020, pp. 185–197. International Committee on Computational Linguistics (2020)
9. Ji, S., Cambria, E., Marttinen, P.: Dilated convolutional attention network for medical code assignment from clinical text. arXiv preprint [arXiv:2009.14578](https://arxiv.org/abs/2009.14578) (2020)
10. Johnson, A.E., et al.: MIMIC-III, a freely accessible critical care database. Sci. Data **3**(1), 1–9 (2016)
11. Jouhet, V., et al.: Automated classification of free-text pathology reports for registration of incident cases of cancer. Methods Inf. Med. **51**(3), 242 (2012)
12. Kavuluru, R., Rios, A., Lu, Y.: An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artif. Intell. Med. **65**(2), 155–166 (2015)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017. [OpenReview.net](https://openreview.net) (2017)
15. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: graph neural networks meet personalized PageRank. In: ICLR 2019. [OpenReview.net](https://openreview.net) (2019)
16. Langville, A.N., Meyer, C.D.: Survey: deeper inside PageRank. Internet Math. **1**(3), 335–380 (2003)
17. Li, F., Yu, H.: ICD coding from clinical text using multi-filter residual convolutional neural network. In: AAAI, vol. 34, pp. 8180–8187 (2020)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) ICLR 2013 (2013)
19. Mullenbach, J., Wiegrefe, S., Duke, J., Sun, J., Eisenstein, J.: Explainable prediction of medical codes from clinical text. In: ACL, June 2018
20. Nadathur, S.G.: Maximising the value of hospital administrative datasets. Aust. Health Rev. **34**(2), 216–223 (2010)
21. Niwattanakul, S., Singthongchai, J., Naenudorn, E., Wanapu, S.: Using of Jaccard coefficient for keywords similarity. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, vol. 1, pp. 380–384 (2013)
22. Ogura, Y., Kobayashi, I.: Text classification based on the latent topics of important sentences extracted by the PageRank algorithm. In: ACL 2013, pp. 46–51. ACL (2013)

23. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
24. Perotte, A.J., Pivovarov, R., Natarajan, K., Weiskopf, N.G., Wood, F.D., Elhadad, N.: Diagnosis code assignment: models and evaluation metrics. *J. Am. Med. Inform. Assoc.* **21**(2), 231–237 (2014)
25. Peyré, G.: Entropic approximation of Wasserstein gradient flows. *SIAM J. Imaging Sci.* **8**(4), 2323–2351 (2015)
26. dos Santos, C.N., Tan, M., Xiang, B., Zhou, B.: Attentive pooling networks. *CoRR abs/1602.03609* (2016)
27. Scozzafava, F., Maru, M., Brignone, F., Torrissi, G., Navigli, R.: Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In: Çelikyilmaz, A., Wen, T. (eds.) *ACL 2020*, pp. 37–46. *ACL* (2020)
28. Shi, H., Xie, P., Hu, Z., Zhang, M., Xing, E.P.: Towards automated ICD coding using deep learning. *arXiv preprint [arXiv:1711.04075](https://arxiv.org/abs/1711.04075)* (2017)
29. Shiv, V., Quirk, C.: Novel positional encodings to enable tree-based transformers. *Adv. Neural Inf. Process. Syst.* **32**, 12081–12091 (2019)
30. Vu, T., Nguyen, D.Q., Nguyen, A.: A label attention model for ICD coding from clinical text. *CoRR abs/2007.06351* (2020). <https://arxiv.org/abs/2007.06351>
31. Wang, S., et al.: Coding electronic health records with adversarial reinforcement path generation. In: Huang, J., Chang, Y., et al. (eds.) *SIGIR 2020*, pp. 801–810. *ACM* (2020)
32. Xie, P., Xing, E.: A neural architecture for automated ICD coding. In: *ACL*, July 2018
33. Xie, X., Xiong, Y., Yu, P.S., Zhu, Y.: EHR coding with multi-scale feature attention and structured knowledge graph propagation. In: Zhu, W., et al. (eds.) *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019*
34. Xie, X., Xiong, Y., Yu, P.S., Zhu, Y.: EHR coding with multi-scale feature attention and structured knowledge graph propagation. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 649–658 (2019)
35. Xu, K., et al.: Multimodal machine learning for automated ICD coding. In: Doshi-Velez, F., et al. (eds.) *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2019, 9–10 August 2019, Ann Arbor, Michigan, USA. Proceedings of Machine Learning Research*, vol. 106, pp. 197–215. *PMLR* (2019)
36. Yin, W., Schütze, H.: Attentive convolution. *CoRR abs/1710.00519* (2017)