Neural Computing and Applications (2023) 35:17633–17644
https://doi.org/10.1007/s00521-023-08581-2

ORIGINAL ARTICLE

Knowledge guided multi-filter residual convolutional neural network for ICD coding from clinical text

Zeyd Boukhers[1,2,4] · Prantik Goswami[1] · Jan Jürjens[1,3]

Received: 30 August 2022 / Accepted: 5 April 2023 / Published online: 17 May 2023

Abstract
A common challenge encountered when using Deep Neural Network models for automatic ICD coding is their potential inability to effectively handle unseen clinical texts, especially when these models are only trained on a limited number of examples. This is because these models rely solely on the patterns and relationships present in the training data, and may not be able to effectively incorporate additional knowledge about the relationships between medical entities. To address this issue, we introduce *KG-MultiResCNN—Knowledge Guided Multi-filter Residual Convolutional Neural Network* model, which combines training examples with external knowledge from the Wikidata Knowledge Graph (KG) in order to better capture the relationships between medical entities. The KG is a structured database that contains a wealth of information about various entities, including medical concepts and their relationships with one another. By incorporating this external knowledge into our model, we are able to improve its ability to predict ICD codes for new clinical texts. In our experiments with the MIMIC-III dataset, we found that the KG-MultiResCNN model significantly outperformed the baseline approaches. This demonstrates the effectiveness of using external knowledge, in addition to training examples, to improve the performance of deep learning models for automatic ICD coding.

Keywords Automatic ICD Coding · Computer-Aided Diagnosis · Knowledge Guided Convolutional Neural Networks · Medical Entity Relationships · Embedding Knowledge Graphs

1 Introduction

In the past decade, Deep Learning (DL) and Natural Language Processing (NLP) techniques have been widely used in healthcare research [1–7] due to a large amount of health data available. One significant application of these techniques is in medical diagnostic decision-making [8, 9], as deep learning approaches applied to medical images have already achieved accuracy on par with human professionals. DL techniques applied to textual data, such as Electronic Health Records (EHR), are also gaining attention, particularly for the automatic detection and assignment of International Classification of Diseases (ICD) codes. The ICD is a globally recognized list of codes developed and maintained by the World Health Organization (WHO) to represent diagnoses and medical procedures with universal codes for healthcare systems such as hospitals and health insurance companies. It is commonly used by healthcare providers for a variety of purposes, including improving the usability and maintainability of records, facilitating reimbursement, and enabling the storage and retrieval of diagnostic and procedural information whenever needed [10, 11]. As part of hospital services, clinical EHRs are often linked to the corresponding ICD codes for each patient's hospital admission, allowing for better organization and management of patient data.

✉ Zeyd Boukhers
zeyd.boukhers@fit.fraunhofer.de

Prantik Goswami
prantik@uni-koblenz.de

Jan Jürjens
juerjens@uni-koblenz.de

[1] University of Koblenz-Landau, 56070 Koblenz, Germany

[2] Fraunhofer Institute for Applied Information Technology FIT, 53757 Sankt Augustin, Germany

[3] Fraunhofer Institute for Software and Systems Engineering ISST, 44227 Dortmund, Germany

[4] Faculty of Medicine and University Hospital Cologne, University of Cologne, 50937 Cologne, Germany

The use of automatic ICD coding from textual clinical notes has been a topic of research for over two decades [12, 13]. Early methods often relied on handcrafted features [14], but as technology and data processing power have improved, a range of approaches have been developed. Perotte et al. [15] used a Support Vector Machine (SVM) to classify "flat" and "hierarchical" ICD codes, while Koopman et al. [16] also used an SVM to classify hierarchical ICD codes related to cancer from textual death certificates. Shi et al. [17] used a character-level Long Short-Term Memory (LSTM) model to identify similarities between discharge summary notes and ICD code descriptions. Prakash et al. [18] developed a neural memory network model called "C-MemNNs" that learned representations from textual data and predicted top-50 and top-100 codes and also used Wikipedia as an external knowledge to improve model performance. Vani et al. [19] created a Grounded Recurrent Neural Network (GRU) that utilized label-specific dimensions for hidden units to predict specific diseases. Baumel et al. [20] used a Hierarchical Attention-Bidirectional Gated Recurrent Unit (HA-GRU) to assign multiple ICD codes to patients' discharge summary notes. Wang et al. [21] proposed a mixed embedding model that calculated the cosine similarity between word embedding vectors and label vectors in the same embedding space to predict the labels.

Li and Yu [22] recently proposed the Multi-Filter Residual Convolutional Neural Network (MultiResCNN) as a state-of-the-art model for predicting multiple possible ICD codes from the content of the discharge summaries. The model uses multiple filter CNN networks followed by residual networks and was evaluated on the MIMIC-III discharge summary notes dataset, where it achieved satisfactory results. However, like many other existing approaches, the model still struggles to effectively capture the correlation between diseases (represented by ICD codes) and the physiological and symptom attributes mentioned in clinical text. This is a significant challenge because most current methods rely only on training examples (i.e., clinical cases documented in clinical texts) to learn this correlation. However, the high dimensionality and sparsity of the feature/class space make it difficult to find a sufficient number of training examples, in reality, to accurately model this relationship. The dimensionality refers to the number of possible diseases and physiological, symptom, and lab-test attributes, while sparsity refers to the rarity of certain attributes in clinical cases. As a result, there is a need for more effective methods that can better handle the high dimensionality and sparsity of this task, and further improve the accuracy of automatic ICD coding from free-text clinical notes.

The goal of this research is to improve the state-of-the-art method for automatic ICD coding from clinical texts, which currently struggles to effectively capture the relationship between diseases and physiological and symptom attributes mentioned in the text. To address this issue, the proposed approach simulates the way physicians interpret clinical texts into diagnoses, using their medical knowledge to understand the clinical situation and the relationships between different diseases, symptoms, and treatments. Consequently, this approach aims to improve the performance of automatic ICD coding by incorporating external medical knowledge in the form of a knowledge graph. To this end, this work enhances the state-of-the-art method proposed by Li and Yu [22] by guiding the model with external medical knowledge. To incorporate this structured medical knowledge into the model, we introduce *KG-MultiResCNN—Knowledge G uidedMulti-filter Residual Convolutional Neural Network* model that is guided by an additional embedding vector. This vector is a knowledge graph embedding of medical entities automatically extracted from the clinical text and is concatenated with the word embedding vector. The model is then trained using both the original text word embeddings and the knowledge graph embeddings. We also compute the Term Frequency-Inverse Document Frequency (TF-IDF) value for each word in the clinical text as a weighting factor for the medical entities and use two residual (ResNet) blocks to extract better feature representation due to the large size of the embedding vectors. The assumption is that medical entities that are not synonyms and have similar relationships should have similar embeddings. Overall, this work aims to tackle a single but important research question: "*Does the inclusion of knowledge graph support the process of automatic ICD coding?*". The main contributions of this work are as follows:

- Improved the MultiResCNN [22] model by introducing an additional embedding layer based on a knowledge graph of significant medical entities extracted from the text;
- Used knowledge graph embedding for automatic ICD coding for the first time, to our knowledge;
- Weighted the importance of each word in the text using the Term Frequency-Inverse Document Frequency (TF-IDF) score as a weighting factor;
- Employed two residual (ResNet) blocks to improve feature representation and handle the large size of the embedding vectors;
- Made all implementations publicly available for further research.

The remainder of this paper is organized as follows: In Sect. 2, we review previous research on the topic and discuss the relevant approaches and their strengths and limitations. Section 3 describes our proposed method in detail with all technical details. Section 4 presents the

results of the experimental evaluation of our method, including statistical analyses and comparisons with other approaches. Finally, in Sect. 5, we summarize the main findings of our research and discuss the implications of the results for future work. We also include recommendations for practical applications and directions for future research.

2 Related work

Assigning an ICD code to a free-text EHR document is a challenging and arduous process. It demands expertise in the healthcare field and can be both financially and error-prone. This has led to prolonged research on developing automatic methods to extract ICD codes from clinical notes for over two decades [12, 13]. In this section, we thoroughly review the most critical ICD coding techniques, grouping them into three distinct categories for enhanced comprehension and organization.

2.1 Classical machine learning

Early efforts to assign ICD codes to inpatient episodes have largely relied on handcrafted features [14] and traditional machine learning models. Perotte et al. [15] used a Support Vector Machine (SVM) to classify flat and hierarchical ICD codes, while Koopman et al. [16] employed a similar SVM approach to classifying hierarchical ICD codes related to cancer from free-text death certificates. Ferrao et al. [23] proposed an adaptive data processing method that utilizes structured electronic health record data and is trained by SVM classifiers to predict codes, resulting in F1-measure values around 52%. Zhou et al. [24] proposed a regular expression-based approach to establish a correspondence between unique ICD codes and diagnosis descriptions in both outpatient and inpatient settings. Diao et al. [25] evaluated the performance of two feature engineering methods for processing discharge diagnosis and procedure texts, using the gradient boosting algorithm on a dataset of 71,709 admissions at Fuwai Hospital and 168 primary diagnoses with ICD-10 codes.

2.2 Neural network-based approaches

Over the past decade, the majority of proposed ICD coding solutions have been based on Neural Networks, such as in [17, 19, 22], due to their impressive performance across a variety of tasks. Shi and colleagues ([17]) utilized character-level LSTM to identify similarities between discharge summary notes and ICD code descriptions. Vani et al. [19] developed a Grounded Recurrent Neural Network (GRU) that incorporates label-specific dimensions for hidden units to predict specific diseases. Baumel et al. [20] employed a Hierarchical Attention-bidirectional Gated Recurrent Unit (HA-GRU) to assign multiple ICD codes to patients' discharge summary notes. Wang et al. [21] proposed a mixed embedding model, assuming that projecting word and label vectors in the same embedding vector space would lead to better results. Their model calculates the cosine similarity between word embedding vectors and label vectors to predict the labels. Xu et al. [26] proposed an ensemble-based approach that combines the outputs of three neural network models, each handling different types of data (unstructured, semi-structured, and tabular). The models utilize CNNs, LSTMs, and decision trees for data processing and classification. The approach was evaluated using MIMIC-III data and demonstrated improved performance by using multiple modalities of data. Meanwhile, Mullenbach et al. [27] proposed the CNN model CAML, which utilizes label attention to enhance ICD coding task performance. The model uses pre-trained word vectors and was tested on MIMIC-III and MIMIC-II discharge summary notes, outperforming previous methods.

As the most recent state-of-the-art model, Li and Yu [22] proposed Multi-Filter Residual Convolutional Neural Network (MultiResCNN) which utilizes a one-hot encoded label vector to predict multiple ICD codes related to the discharge summary text. Their approach uses a multiple-filter CNN network, with a residual network [28] following each filter, and employs a label attention mechanism for better prediction accuracy. They evaluated their model on the MIMIC-III discharge summary notes dataset and showed improved performance with both MIMIC-Full codes and MIMIC-50 codes.

The limitation of these approaches is that they rely solely on the examples present in the training set, which can only represent a small subset of the vast and complex space of diseases, symptoms, and epidemiological factors. This can restrict the model's ability to generalize to new and unseen data. To overcome this limitation, it is crucial to incorporate external knowledge sources that can augment the training data and provide additional information to improve the performance of the models.

2.3 Knowledge-enhanced approaches

Many studies have investigated the effect of external information sources on medical text understanding [18, 29, 30]. While Kumar Chanda et al. [30] proposed a method for learning medical term embeddings from limited notes by using medical term definitions as external knowledge, Bai and Vucetic [31] built upon the CAML model by incorporating a Knowledge Source Integration (KSI) framework to improve performance. KSI uses superficial knowledge from Wikipedia to add extra weight to the input text for ICD code prediction, specifically

focusing on rare diseases. The model was evaluated on the MIMIC-III dataset and showed improved performance in predicting rare diseases. These studies demonstrated the need for external knowledge, but the unstructured knowledge used can be difficult for the machine to process. As an alternative, it may be beneficial to incorporate structured knowledge sources in the form of knowledge graphs.

Choi et al. [32] introduced GRAM, which combines information from medical ontologies with deep learning models via attention mechanism. Ancestors of less frequent medical concepts are adaptively combined by frequency and attention, and the attention mechanism is trained end-to-end. This means that if enough training data are available, GRAM achieves comparable results without incorporating the medical ontology. In contrast, KAME [33] exploits a medical ontology (i.e., ICD 9) to learn representations of medical codes and their ancestors in the whole prediction process. Bao et al. [34] used ICD descriptions as external knowledge sources to improve medical code prediction in their hybrid capsule network model with a bi-directional LSTM and label embedding framework. Similarly, Du et al. [35] used GCN to obtain diagnosis codes' semantic representations and construct a co-occurrence graph from EHR data, improving token extraction with an attention mechanism to model the interaction between diagnosis codes' ontology representations and clinical notes. Peng et al. [36] proposed MIPO, a healthcare representation learning model that uses medical knowledge and patient journey to predict future diagnoses. MIPO consists of a task-specific representation learning module and a graph-embedding module, and it jointly learns task-specific and ontology-based objectives.

The works mentioned above utilize structured knowledge in the entire prediction process, however, the medical ontologies and ICD descriptions predominantly used primarily reveal connections among diseases and not all medical entities mentioned in medical texts, such as symptoms and epidemiological factors. This can hinder the machine's ability to effectively utilize all available medical information and evidence-based knowledge during the prediction process. To address this limitation, a more comprehensive knowledge graph should be properly integrated, which can enable the machine to incorporate a broader range of information and improve the accuracy of predictions.

3 KG-MultiResCNN

This paper presents a novel model called *KG-MultiResCNN—Knowledge Guided Multi-filter Residual Convolutional Neural Network*, based on the state-of-the-art approach proposed by Li and Yu [22]. The main contribution of this work is to predict disease ICD codes from unstructured clinical texts by leveraging a knowledge graph. The model first extracts tokens from the clinical text and represents them numerically, weighting them according to their importance. Subsequently, it identifies medical entities and represents the relationships between them numerically using knowledge graph embedding. As illustrated in Fig. 1, these representations are concatenated and passed through a Multi-filter Residual Convolutional Neural Network to predict the ICD code. We employed CNNs due to their effectiveness in processing sequential unstructured data such as free text. Due to the complexity of the task, a deep CNN is needed. Therefore, residual blocks have been considered to address the vanishing gradient problem. In the following, we discuss each of the elements of *KG-MultiResCNN*:

3.1 Word embedding input

The first part of the input layer is an embedding matrix (E) obtained from the sequence of the words of the text document. The word sequence is denoted as $\mathbf{w}$, which is defined as $\mathbf{w} = (w_1, w_2, ..., w_n)$, where n is the total number of words present in the text. For each word, the embedding vector is obtained using the pretrained word2-vec model [37]. Furthermore, each word embedding is weighted using a TF-IDF[1] score. TF-IDF measures the relevance of words such that those frequent in the document but rare in the collection are considered most relevant. Specifically, the embedding vector can be formulated as $\mathbf{e} = g \times \mathbf{u}$ where $\mathbf{u}$ is the word embedding and $g > 0$ is the TF-IDF score of that word. Consequently, the the word embedding input part becomes $E = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\}$ where $e_i \in \mathbb{R}^{d^{(w)}}$. $d^{(w)}$ is the dimension of the word embedding vector.

3.2 Input KG-embedding input

The second part of the input layer is the knowledge graph embedding matrix (K), which encode the relationships between the medical entities present in the clinical text with all related entities regardless of whether they are present in the clinical text or not. To this end, we extract from $\mathbf{w}$ the most significant medical entities using a domain-specific Named Entity Recognition model .[2] This results in the sequence $\mathbf{t}$ denoted as $\mathbf{t} = \{t_1, t_2, ..., t_m\}$, where m is the number of medically significant entities extracted by the entity extraction model. Using each entity j_j, a Knowledge Graph is queried to obtain the knowledge graph embedding k_j. Hence the knowledge graph

[1] Term frequency-inverse document frequency.

[2] https://huggingface.co/samrawal/bert-base-uncased_clinical-ner.

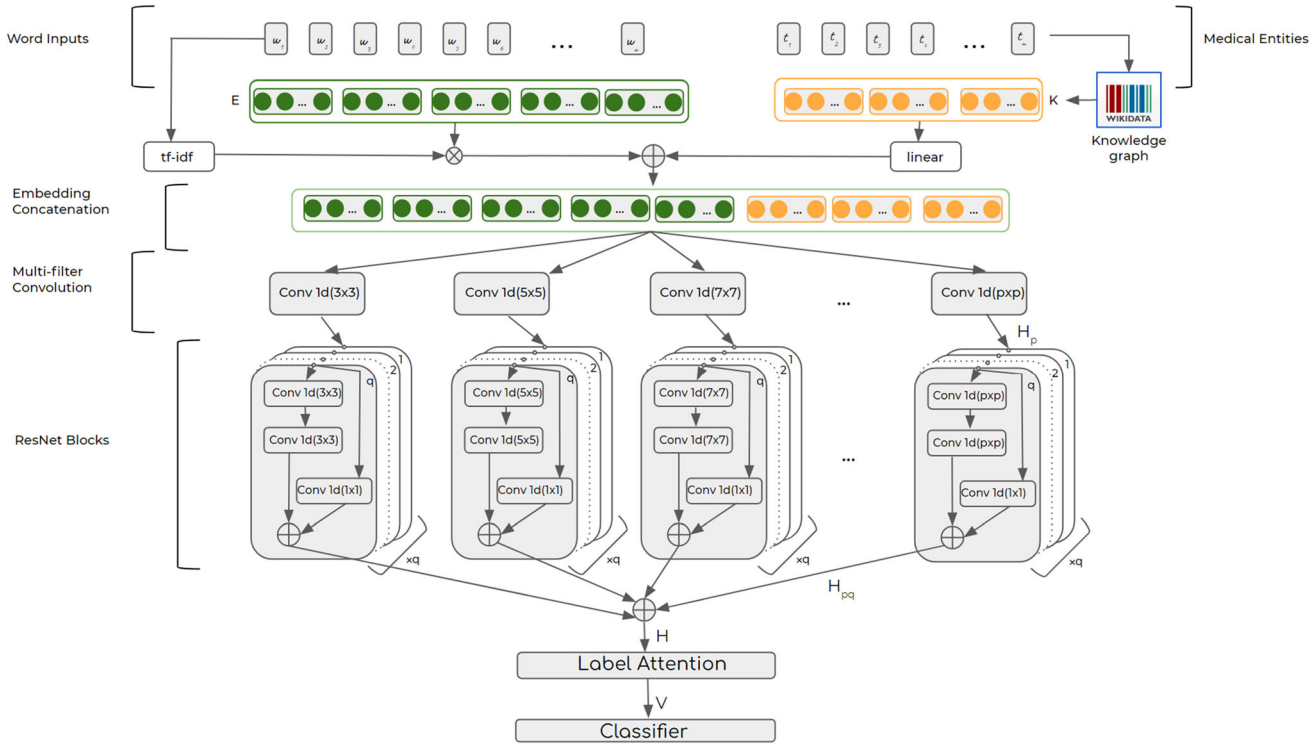

Fig. 1 An overview of "*Kg-MultiResCNN*" architecture for ICD code prediction using a Multi-filter Residual Convolutional Neural Network

embedding matrix becomes, $K = \{k_1, k_2, ..., k_m\} \in \mathbb{R}^{m \times d^{(k)}}$, where $d^{(k)}$ denotes the dimension of the knowledge embedding. In this paper, we employed PyTorch BigGraph (PGB)[38] which is an embedding system provided by Meta Research[3] community. PGB learns the node and edges representations of massive knowledge graphs and embeds the nodes and relations in the graph. Its strength lies in the fact that it is trained on the large Wikidata[4] knowledge graph with 78 million entities and 4131 relations and provides embedding of 200 dimensions. It is highly likely that the medical entities extracted from the clinical text exist in Wikidata and are connected to other medical entities with several relationship types. The word embedding matrix and the KG embedding matrix jointly serve as the input layer (i.e., clinical text representation) to the model.

3.3 Multi-filter convolution layer

To map the clinical text representation to the ICD codes, we followed the work of Li and Yu [22] by building a multi-filter 1-dimensional Convolutional Neural Network architecture. The strategy is to pass the varied length of texts through a parallel set of CNN networks. However, the kernel size is of different lengths for each CNN filter. Given p filters, the corresponding kernel size would be k_p and the convolution filter would be $W_p \in \mathbb{R}^{k_p \times d^{(e)} \times d^{(c)}}$ where $d^{(e)}$ is the input dimension and $d^{(c)}$ is the output dimension. In general, the filter/convolution operation on a vector reduces the size of the output vector. However, in this approach, we aim to keep the size of the output vector the same as the input. To this end, the number of parameters is calculated as follows:

$$L_{\text{out}} = \left[\frac{L_{\text{in}} + 2 \times \text{padding} - \text{dilation} \times (\text{kernel\_size} - 1) - 1}{\text{stride}} + 1\right]$$

By setting the stride = 1, dilation = 1, kernel_size = k, and padding = $floor(\frac{k}{2})$, we can achieve our goal of same output size. With all these adjustments, the 1-Dimensional convolution operation can be formalized as:

$$\mathsf{C}_{p,j}(E) = W_p^T \otimes E^{j:j+k_p-1}$$

$$H_p = \sum_{j=1}^{n} \tanh(\mathsf{C}_{p,j}(E))$$

Here, $\otimes$ represents a convolution operation and $\mathsf{C}_{p,j}$ indicates the output of p^{th} convolution where the input matrix position starts from j *th* row and ends at the row $j + k_p - 1$. H_n indicates the final layer output after the convolution output is passed though *tanh* activation for total n sequence of input and then concatenated (indicated by $\sum$) together.

[3] https://github.com/facebookresearch.

[4] https://www.wikidata.org.

3.4 Residual convolution layer

The output of each convolutional filter again goes through a series of convolution filters called a residual block. Each of these blocks consists of 3 convolution layers. A typical 1-D convolution architecture is shown in Fig. 2, where the convolution filter W_p slides through the embedding matrix E with a stride of 1. Formally, if we consider p multi-filter convolution layers then each of these convolution filters has a series of q residual blocks on top. Each of the residual blocks have three convolution filters, namely $r_{pq_1}, r_{pq_2}, r_{pq_3}$ and their corresponding filter weights are $W_{pq_1}, W_{pq_2}, W_{pq_3}$, where r_{pq} is the q^{th} residual block on top of p^{th} multi-filter convolution layer. The output of each convolution filter inside a residual block can be formulated as

$$\complement_{pq_1,j}(X) = W_{pq_1}^T \otimes X^{j:j+k_{pq_1}-1},$$

$$H_{pq_1} = \sum_{j=1}^{n} tanh(\complement_{pq_1,j}(X)),$$

$$H_{pq_2} = \sum_{j=1}^{n} \complement_{pq_2,j}(H_{pq_1}),$$

$$H_{pq_3} = \sum_{j=1}^{n} \complement_{pq_3,j}(X),$$

$$H_{pq} = tanh(H_{pq_2} + H_{pq_3}),$$

where $+$ represents the element-wise addition and H_{pq} is the final output from the q^{th} residual block that used the initial input matrix from the output of p^{th} multi-filter convolutional block. X is the input matrix to each of the residual blocks. The first residual block is fed with the output of the multi-filter convolution layer. Finally, the output of each of the final residual blocks is concatenated together to use in the next step. The final output can be formulated as:

$$H = \sum_{1}^{p} H_{pq}$$

where p is the total no of filters used in the multi-filter convolution layer.

3.5 Attention layer

The final output matrix H is typically reduced to a vector using the max-pooling operation before passing it to a classifier. However, in this model, we used an additional label attention step as suggested by Mullenbach et al. [27]. The idea is that some words have higher weights for a label for multi-class classification. Therefore, the label attention can select the most relevant k-grams from the text that can benefit in predicting the correct label. Formally, the procedure is to create a vector parameter U for the labels and then compute the matrix–vector product HU. Then we use a softmax layer to obtain the word distribution in the text.

$$\alpha = \text{softmax}(HU)$$

where α is the attention vector. To get the final vector representation from the attention layer we again perform a

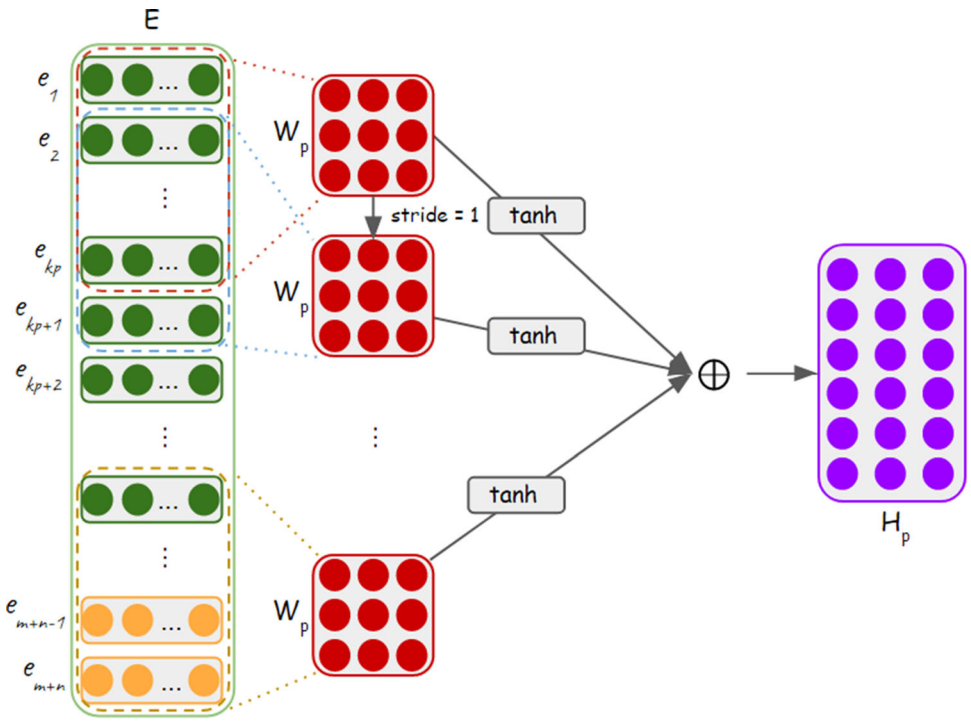

Fig. 2 A general architectural overview of 1-D convolution with stride 1

matrix multiplication between the attention vector α and the input matrix H. The final output is formulated as

$$V = \alpha^T H$$

3.6 Output layer

The output layer is a superficial linear layer that takes the input V from the attention layer. The score vector of all the labels is obtained using the sum-pooling operation on the output vector resulting from a linear transformation. The final probability vector is calculated using sigmoid activation on the score vector for multi-class classification, such that $Y = VW$, where W of dimension $((p \times d^{pq}), l)$ is the weight matrix. Here, p is the total number of convolution filters used in the multi-filter convolution step, and d^{pq} is the output dimension from the residual convolution layer. l is the output dimension, the total number of labels that we are classifying. The score vector $\hat{Y}$ can be formulated as:

$$\hat{Y} = \text{pooling}\left(\sum_{j=1}^{l} Y_{ij}\right)$$

and the final predicted vector is:

$$\tilde{Y} = \sigma(\hat{Y})$$

4 Results

In this section, we evaluate the effectiveness of the *KG-MultiResCNN* against the baseline state-of-the-art approaches. To reproduce the results and further improvements, we made the implementation of the *KG-MultiResCNN* publicly available [5] and the details of the architecture are illustrated in Fig. 3.

We conducted several experiments with different parameters to determine the optimal operation settings for our model. We found that using 100-dimensional embedding vectors for the input word embedding yielded better performance than using higher-dimensional embedding vectors. Additionally, the number of words in the clinical text played a significant role in the model's performance, with a maximum of 3000 words resulting in the best performance. We also discovered that using a maximum of 30 medical entities extracted from the clinical text led to optimal performance, and architecture with nine CNN channels was the most advantageous for modeling this number of words. For the combined input of word embeddings and KG embeddings, the model performed best with two residual layers. Although the complexity of the "KG-MultiResCNN" model was relatively high, it had comparable computational costs to the "MultiResCNN" model. However, if the number of words and extracted medical entities is higher, more CNN channels and/or residual layers would be needed, leading to increased computational costs.

4.1 Dataset

Medical Information Mart for Intensive Care (MIMIC-III) [39] is one of the largest labeled datasets of clinical texts with clinical records of around 40 thousand patients. Also, it is used by most of the state-of-the-art approaches [22, 26, 27, 40, 41]. Therefore, MIMIC-III is adopted in this work to be the evaluation dataset. Similarly to Mullenbach et al. [27] and Li and Yu [22], we use in this work the "Discharge summaries" which contain a general description of the patient, starting from their medical history to the final discharge notes. On top of that, we aim also to assess the capability of *KG-MultiResCNN* on predicting the ICD codes from the *clinical descriptive texts* and without using the discharge notes. We mean by *clinical descriptive texts*, texts that describe the clinical case (e.g., lab tests and clinical observations) without any explicit or implicit clue of the diagnosis and they include clinical notes, nursery observations and free-text notes from medical examinations such as radiology, electrocardiography, echocardiography, and respiratory check examinations. Following the baseline approaches (e.g., [22]), we consider two experiments, one will full codes (4216) and the second one with the top occurring 50-codes. This means that only clinical instances, that are assigned to at least one of the top 50 most frequent codes, are considered. This is because most of the ICD codes are assigned to very few hospital admissions.

4.2 Evaluation metrics

KG-MultiResCNN is a multi-class classifier, distinguishing between several ICD codes. It is customary to evaluate this kind of classifier at a range of thresholds $p_\tau \in [0; 1]$ for the decision $p > p_\tau$ and then represent the results in the form of Receiver Operating Characteristic (ROC) curves and Area Under ROC (AUROC). However, although the distinction is important, it may not properly address clinical usefulness [42–47]. More specifically, a false negative prediction is more harmful than a false positive decision. In that case, a model with high sensitivity may be preferable to a model with high specificity and low sensitivity. In other words, a model is clinically useful if its decisions for patients lead to a better ratio between benefits and harms compared to not using the model. Therefore, we employed other evaluation

[5] https://KG-MultiResCNN.ai-research.net.

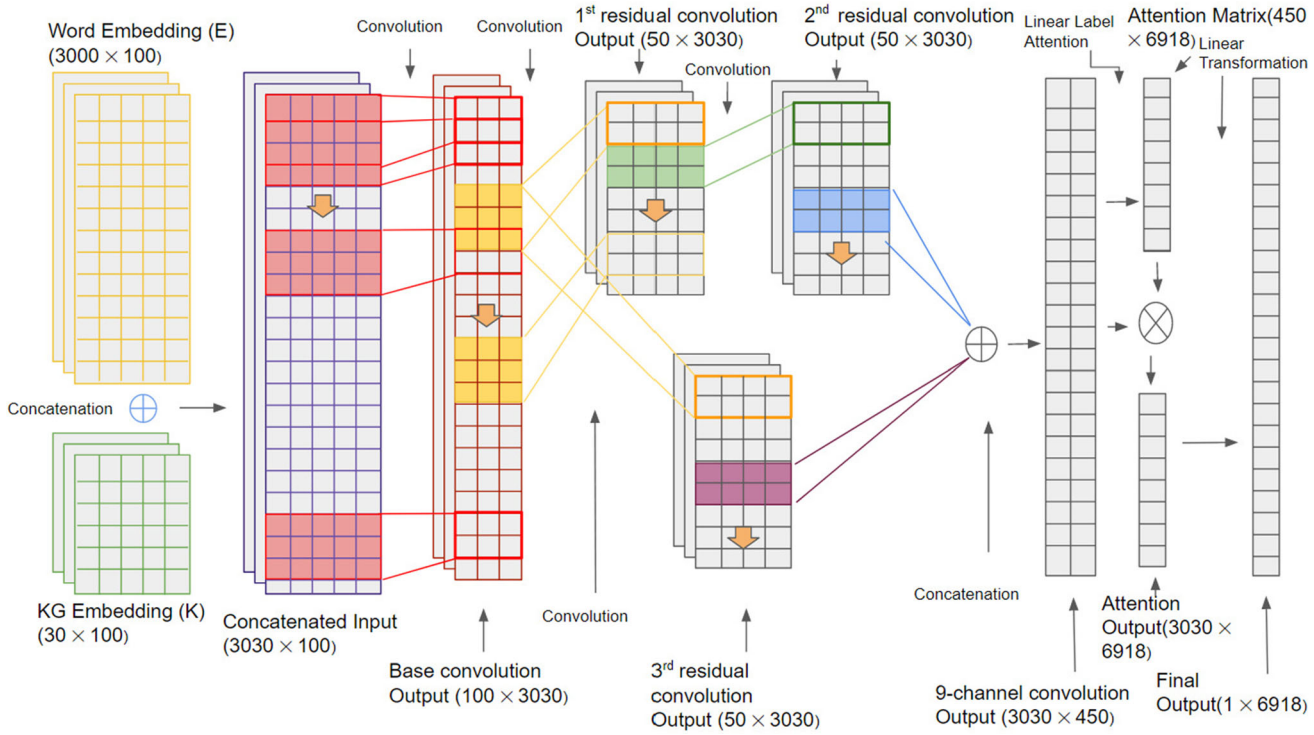

Fig. 3 Full implemented architecture of "KG-MultiResCNN."

metrics: AUC, Precision@5 (P@5), Precision@8 (P@8), and Precision@15 (P@15). Since the classes (ICD codes) are not supposed to be balanced, micro and macro averaging are adopted for better computation of the average score among the different classes.

4.3 Baselines

Because the main contribution of *KG-MultiResCNN* is enhancing *MultiResCNN* [22] with external knowledge guidance, the main comparison is against *MultiResCNN*. In addition, we consider the following baselines:

- **Logistic regression** (LR): Mullenbach et al. [27] used Logistic Regression (LR) to predict ICD codes using a unigram bag-of-words vector for all words in the MIMIC-III text data.
- **SVM**: Perotte et al. [15] experimented with hierarchical and flat ICD code prediction on MIMIC-II using Support Vector Machine (SVM). Later, Xie et al. [48] used also SVM for hierarchical ICD code prediction on the MIMIC-III dataset. Their model performed moderately with 10,000 unigram word vectors and with TF-IDF weighting.
- **CNN**: Mullenbach et al. [27] experimented with the performance of 1D-CNN on classifying ICD codes from MIMIC-III clinical notes.
- **Bi-GRU**: Mullenbach et al. [27] achieved modest performance by applying the Bi-GRU [49] for ICD classification with MIMIC-III clinical notes.
- **C-LSTM-Att**: Shi et al. [17] used an LSTM based language model called the Character-aware LSTM-based Attention (C-LSTM-Att). The model used an attention mechanism to handle the mismatch between notes and ICD codes and was used to predict the top 50 ICD codes from the MIMIC-III dataset.
- **LEAM**: Wang et al. [21] proposed a text classification model called the Label Embedding Attentive Model (LEAM) that predicts the top 50 ICD codes from the MIMIC-III dataset. The model projects the embedding of words and labels in the same latent vector space and calculates the similarities between the embeddings.
- **CAML**: Mullenbach et al. [27] introduced the Convolutional Attention Network for Multi-Label classification applied on ICD code classification using MIMIC-III notes. The model achieved high performance for multi-label ICD code classification.
- **DR-CAML**: As an extension of CAML, Mullenbach et al. [27] introduced the Description Regularized CAML. The model used the text description of the codes for better prediction accuracy.

Table 1 Comparison results of KG-MultiResCNN against the baseline methods on predicting ICD codes using "Discharge summary" notes in terms of F1-Score

Model	Full codes		Top-50 codes	
	Micro(%)		Macro(%)	
LR	27.2	1.1	53.3	47.7
Flat SVM	39.7	–	–	–
Hierarchy SVM	44.1	–	–	–
C-LSTM-Att	–	–	53.2	–
CNN	41.9	4.2	62.5	57.6
Bi-GRU	41.7	3.8	54.9	48.4
LEAM	–	–	61.9	54.0
CAML	53.9	8.8	61.4	53.2
DR-CAML	52.9	8.6	63.3	57.6
MultiResCNN	55.2	8.5	67.0	60.6
KG-MultiResCNN	**56.1**±0.1	**10.2** ± 0.1	**69.5** ± 0.1	**64.5** ±0.1

Bold values signify the highest-performing methods among those compared

± indicates standard deviations

4.4 Comparison against the baselines

In this comparison, only "Discharge summary" is considered because it is the only type of note used by the baselines. As the main comparison, we compared KG-MultiResCNN against all baseline approaches mentioned above. Table 1 presents the comparative results in terms of Micro and Macro F1-score averages for both "full-codes" and "50 codes" experiments. It is evident from the results that KG-MultiResCNN significantly outperforms all the baseline approaches, including current state-of-the-art MultiResCNN. Even with the full diagnosis and procedural ICD coding setting, KG-MultiResCNN acquired a Micro F1-score average of 56.1%, surpassing all approaches.

For further evaluation, we compare KG-MultiResCNN against MultiResCNN in terms of predicting the diagnosis ICD codes using the "Discharge summary" notes. Table 2 shows the comparison results between the two approaches, demonstrating that KG-MultiResCNN achieved better macro and micro F1-score compared to MultiResCNN. It is important to note that the results of MultiResCNN can be slightly different than what was mentioned on the paper [22] as we reproduced them to guarantee a fair comparison. When applied to the "full code" dataset, the guidance of the knowledge graph in KG-MultiResCNN improved the Micro F1-score average by 0.9%. In terms of Macro F1-score average, KG-MultiResCNN is better with 1.7%. Similarly, for the "50-code" dataset, "KG-MultiResCNN" achieved better results compared to MultiResCNN, where the Micro F1-score and Macro F1-score are improved with 1.46% and 3.9%, respectively. The results also show a stable standard deviation for both the "full-codes" and "50 codes" experiments. Despite the result improvement is marginal, it clearly answers the research question raised in this work and proves that guiding the model with medical knowledge graph embeddings of clinical entities is beneficial in automatic ICD coding.

4.5 Results on different note types

Since all the baseline approaches used only "discharge summary" notes which might explicitly comprise the disease, we aim to evaluate the performance of KG-MultiResCNN on the other note types that definitely do not contain an explicit indication of the disease.

Table 3 illustrates a comparative results of "KG-MultiResCNN" with different notes combination for the full code prediction and for top 50 code prediction settings. As anticipated, the model performed better when using only "Discharge summary" notes. By including "Physician" and "Nursing" notes, the results drop slightly, which can be explained by the high dimensionality of the input layer and the complex relationships between the huge number of entities in the text. We assume that a more sophisticated architecture with more layers would work better with a large number of tokens/entities. Another reason could be the huge amount of indirect or irrelevant information that

Table 2 Comparison results of KG-MultiResCNN against KG-MultiResCNN on diagnosis ICD code with "Discharge summary" notes

Model	Full codes				Top-50 codes				P@8	P@15	P@5
	Micro (%)		Macro (%)		Micro (%)		Macro (%)				
	F1	AUC	F1	AUC	F1	AUC	F1	AUC			
MultiResCNN	55.2	**98.6**	8.5	**90.5**	67.0	94.5	60.6	92.5	59.1	43.7	57.5
KG-MultiResCNN	**56.1** ±0.1	98.4	**10.2** ± 0.1	87.1	**69.5** ±0.1	**94.5**	**64.5** ± 0.1	**92.7**	**59.9**	**44.0**	**57.8**

Bold values signify the highest-performing methods among those compared

± indicates standard deviations

Table 3 Comparison results of KG-MultiResCNN on full and top 50 diagnosis ICD codes with multiple note combinations

Notes type	Full codes				Top-50 codes				P@8	P@15	P@5
	Micro (%)		Macro (%)		Micro (%)		Macro (%)				
	F1	AUC	F1	AUC	F1	AUC	F1	AUC			
"Discharge summary" notes	**53.8**	98.4	**10.2**	87.1	**69.06**	94.5	**64.21**	92.7	59.9	44.0	57.8
"Discharge summary" +"Nursing" +"Physician" notes	53.4	98.1	8.8	85.3	68.19	93.7	61.83	91.5	58.9	43.3	57.4
"Nursing" +"Physician" notes	30.5	93.2	2.46	71.5	48.32	82.7	38.13	77.2	42.2	30.7	46.8

Bold values signify the highest-performing methods among those compared

Table 4 Performance comparison between *KG-MultiResCNN* and *MultiResCNN*

	MultiResCNN	KG-MultiResCNN
Trainable Parameters (million)	11.9	11.9
Training Time (seconds/epoch)	1026	2185
Number of epochs	26	15

misleads the model. Due to the same reasons, the performance drops significantly when using only "Physician" and "Nursing" notes. However, the results are still promising for using the model in other tasks (e.g., preliminary diagnosis) and/or for further improvements of the model.

4.6 Performance

Table 4 presents the performance comparison between *KG-MultiResCNN* and the state-of-the-art baseline *MultiResCNN* from different aspects. As shown in the table, *KG-MultiResCNN* converges after 15 epochs only, whereas *MultiResCNN* took 26 epochs to converge. Also, both models have the same number of training parameters. However, *KG-MultiResCNN* takes about 2185 s for each epoch whereas, *MultiResCNN* takes about half of the time. This is due to the higher complexity of *KG-MultiResCNN*. For instance, *KG-MultiResCNN* uses nine convolution channels compared to *MultiResCNN* which uses only six.

5 Conclusion

In this study, we presented KG-MultiResCNN, a Multi-filter Residual Convolutional Neural Network model for predicting multi-label ICD codes using clinical text embeddings. KG-MultiResCNN incorporates medical knowledge graph embeddings that capture the relationships between medical entities in the clinical text. It also considers the relevance of each word by weighting its embedding with a TF-IDF score based on its occurrence in the document and corpus. The obtained results demonstrate that KG-MultiResCNN outperforms state-of-the-art methods, especially with discharge summary notes, which provide critical patient information.

Future research will focus on constructing a medical-specific knowledge graph to address the limitations of the currently adopted knowledge graph, which contains irrelevant relationships. This new graph will be automatically generated from unstructured medical sources like Wikipedia articles and scientific papers. We also plan to combine knowledge representation (via a knowledge graph) with concept representation (via an ontology) to create a model capable of understanding data at three levels: examples from training data, knowledge from the knowledge graph, and the general framework of the data domain.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets generated during and/or analyzed during the current study are available in the MIMIC III repository, http://dx.doi.org/10.13026/C2XW26

Declarations

Conflict of interest The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript, and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

References

1. Wu Y, Jiang M, Lei J, Xu H (2015) Named entity recognition in Chinese clinical text using deep neural network. Stud Health Technol Inform 216:624
2. Nickerson P, Tighe P, Shickel B, Rashidi P (2016) Deep neural network architectures for forecasting analgesic response. In: 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, pp 2966–2969
3. Nguyen P, Tran T, Wickramasinghe N, Venkatesh S (2016) Deepr: a convolutional net for medical records. IEEE J Biomed Health Inform 21(1):22–30
4. Wickramasinghe N (2017) Deepr: a convolutional net for medical records
5. Fries J A (2016) Brundlefly at semeval-2016 task 12: recurrent neural networks vs. joint inference for clinical temporal information extraction. arXiv preprint arXiv:1606.01433
6. Lv X, Guan Y, Yang J, Wu J (2016) Clinical relation extraction with deep learning. Int J Hybrid Inf Technol 9(7):237–248
7. Liu Y, Ge T, Mathews KS, Ji H, McGuinness D L (2018) Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. arXiv preprint arXiv:1804.04225
8. Lee J-G, Jun S, Cho Y-W, Lee H, Kim GB, Seo JB, Kim N (2017) Deep learning in medical imaging: general overview. Korean J Radiol 18(4):570–584
9. Suzuki K (2017) Overview of deep learning in medical imaging. Radiol Phys Technol 10(3):257–273
10. Bottle A, Aylin P (2008) Intelligent information: a national system for monitoring clinical performance. Health services research 43(1p1), 10–31
11. Nadathur SG (2010) Maximising the value of hospital administrative datasets. Aust Health Rev 34(2):216–223
12. Larkey LS, Croft W B (1996) Combining classifiers in text categorization. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, pp 289–297
13. de Lima L R, Laender AH, Ribeiro-Neto B A (1998) A hierarchical approach to the automatic categorization of medical documents. In: Proceedings of the seventh international conference on information and knowledge management, pp 132–139
14. Scheurwegs E, Luyckx K, Luyten L, Daelemans W, Van den Bulcke T (2016) Data integration of structured and unstructured sources for assigning clinical codes to patient stays. J Am Med Inform Assoc 23(e1):11–19
15. Perotte A, Pivovarov R, Natarajan K, Weiskopf N, Wood F, Elhadad N (2014) Diagnosis code assignment: models and evaluation metrics. J Am Med Inform Assoc 21(2):231–237
16. Koopman B, Zuccon G, Nguyen A, Bergheim A, Grayson N (2015) Automatic icd-10 classification of cancers from free-text death certificates. Int J Med Inform 84(11):956–965
17. Shi H, Xie P, Hu Z, Zhang M, Xing E P (2017) Towards automated icd coding using deep learning. arXiv preprint arXiv:1711.04075
18. Prakash A, Zhao S, Hasan S A, Datla V, Lee K, Qadir A, Liu J, Farri O (2017) Condensed memory networks for clinical diagnostic inferencing. In: Thirty-first AAAI conference on artificial intelligence
19. Vani A, Jernite Y, Sontag D (2017) Grounded recurrent neural networks. arXiv preprint arXiv:1705.08557
20. Baumel T, Nassour-Kassis J, Cohen R, Elhadad M, Elhadad N (2018) Multi-label classification of patient notes: case study on icd code assignment. In: Workshops at the thirty-second AAAI conference on artificial intelligence
21. Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. arXiv preprint arXiv:1805.04174
22. Li F, Yu H (2020) Icd coding from clinical text using multi-filter residual convolutional neural network. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, pp 8180–8187
23. Ferrão J C, Janela F, Oliveira MD, Martins H M (2013) Using structured ehr data and svm to support icd-9-cm coding. In: 2013 IEEE international conference on healthcare informatics, IEEE, pp 511–516
24. Zhou L, Cheng C, Ou D, Huang H (2020) Construction of a semi-automatic icd-10 coding system. BMC Med Inform Decision Mak 20(1):1–12
25. Diao X, Huo Y, Zhao S, Yuan J, Cui M, Wang Y, Lian X, Zhao W (2021) Automated icd coding for primary diagnosis via clinically interpretable machine learning. Int J Med Inform 153:104543
26. Xu K, Lam M, Pang J, Gao X, Band C, Mathur P, Papay F, Khanna AK, Cywinski J B, Maheshwari K (2019) Multimodal machine learning for automated icd coding. In: Machine Learning for Healthcare Conference, PMLR, pp 197–215
27. Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J (2018) Explainable prediction of medical codes from clinical text. arXiv preprint arXiv:1802.05695
28. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
29. Bai T, Egleston B L, Bleicher R, Vucetic S (2019) Medical concept representation learning from multi-source data. In: IJCAI: proceedings of the conference, NIH Public Access, vol 2019, p 4897
30. Chanda AK, Bai T, Yang Z, Vucetic S (2022) Improving medical term embeddings using umls metathesaurus. BMC Med Inform Decision Mak 22(1):1–12
31. Bai T, Vucetic S (2019) Improving medical code prediction from clinical text via incorporating online knowledge sources. In: The World Wide Web Conference, pp 72–82
32. Choi E, Bahadori M T, Song L, Stewart W F, Sun J (2017) Gram: graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 787–795
33. Ma F, You Q, Xiao H, Chitta R, Zhou J, Gao J (2018) Kame: knowledge-based attention model for diagnosis prediction in healthcare. In: Proceedings of the 27th ACM international conference on information and knowledge management, pp 743–752
34. Bao W, Lin H, Zhang Y, Wang J, Zhang S (2021) Medical code prediction via capsule networks and icd knowledge. BMC Med Inform Decision Mak 21(2):1–12
35. Du Y, Luo P, Hong X, Xu T, Zhang Z, Ren C, Zheng Y, Chen E (2021) Inheritance-guided hierarchical assignment for clinical automatic diagnosis. In: International conference on database systems for advanced applications, Springer, pp 461–477
36. Peng X, Long G, Shen T, Wang S, Niu Z, Zhang C (2021) Mimo: mutual integration of patient journey and medical ontology for healthcare representation learning. arXiv preprint arXiv:2107.09288
37. Mikolov T, Sutskever I, Chen K, Corrado G S, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp 3111–3119
38. Lerer A, Wu L, Shen J, Lacroix T, Wehrstedt L, Bose A, Peysakhovich A (2019) Pytorch-biggraph: A large-scale graph embedding system. arXiv preprint arXiv:1903.12287
39. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG (2016) Mimic-iii, a freely accessible critical care database. Sci Data 3(1):1–9

40. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H (2021) Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthc (HEALTH) 3(1):1–23
41. Zhang D, Yin C, Zeng J, Yuan X, Zhang P (2020) Combining structured and unstructured data for predictive models: a deep learning approach. BMC Med Inform Decision Mak 20(1):1–11
42. Nugues M, Roberts C (2003) Coral mortality and interaction with algae in relation to sedimentation. Coral Reefs 22(4):507–516
43. McGeechan K, Macaskill P, Irwig L, Bossuyt PM (2014) An assessment of the relationship between clinical utility and predictive ability measures and the impact of mean risk in the population. BMC Med Res Methodol 14(1):1–12
44. Talluri R, Shete S (2016) Using the weighted area under the net benefit curve for decision curve analysis. BMC Med Inform Decision Mak 16(1):1–9
45. Ten Haaf K, Jeon J, Tammemägi MC, Han SS, Kong CY, Plevritis SK, Feuer EJ, de Koning HJ, Steyerberg EW, Meza R (2017) Risk prediction models for selection of lung cancer screening candidates: a retrospective validation study. PLoS Med 14(4):1002277
46. Vickers AJ, Cronin AM (2010) Everything you always wanted to know about evaluating prediction models (but were too afraid to ask). Urology 76(6):1298–1301
47. Steyerberg EW, Vergouwe Y (2014) Towards better clinical prediction models: seven steps for development and an abcd for validation. Eur Heart J 35(29):1925–1931
48. Xie X, Xiong Y, Yu P S, Zhu Y (2019) 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
49. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078