

ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network

Fei Li,¹ Hong Yu^{1,2,3,4}

¹Department of Computer Science, University of Massachusetts Lowell, Lowell, MA, United States

²Center for Healthcare Organization and Implementation Research,
Bedford Veterans Affairs Medical Center, Bedford, MA, United States

³Department of Medicine, University of Massachusetts Medical School, Worcester, MA, United States

⁴School of Computer Science, University of Massachusetts, Amherst, MA, United States
{fei_li, hong_yu}@uml.edu

Abstract

Automated ICD coding, which assigns the International Classification of Disease codes to patient visits, has attracted much research attention since it can save time and labor for billing. The previous state-of-the-art model utilized one convolutional layer to build document representations for predicting ICD codes. However, the lengths and grammar of text fragments, which are closely related to ICD coding, vary a lot in different documents. Therefore, a flat and fixed-length convolutional architecture may not be capable of learning good document representations. In this paper, we proposed a **Multi-Filter Residual Convolutional Neural Network** (MultiResCNN) for ICD coding. The innovations of our model are two-folds: it utilizes a multi-filter convolutional layer to capture various text patterns with different lengths and a residual convolutional layer to enlarge the receptive field. We evaluated the effectiveness of our model on the widely-used MIMIC dataset. On the full code set of MIMIC-III, our model outperformed the state-of-the-art model in 4 out of 6 evaluation metrics. On the top-50 code set of MIMIC-III and the full code set of MIMIC-II, our model outperformed all the existing and state-of-the-art models in all evaluation metrics. The code is available at <https://github.com/foxf823/Multi-Filter-Residual-Convolutional-Neural-Network>.

Introduction

The International Classification of Diseases (ICD), which is organized by the World Health Organization, is a common coding method used in various healthcare systems such as hospitals. It includes many pre-defined ICD codes which can be assigned to patients' files such as electronic health records (EHRs). These codes represent diagnostic and procedural information during patient visits. Healthcare providers and insurance companies need these information to diagnose patients and bill for services (Bottle and Aylin 2008). However, manual ICD coding has been demonstrated to be labor-consuming and costly (O'malley et al. 2005).

The research community has investigated a number of approaches for automated ICD coding, including the models based on both traditional machine learning (Perotte et al. 2013; Kavuluru, Rios, and Lu 2015) and deep learning (Shi

Table 1: Examples of clinical text fragments and their corresponding ICD codes.

998.32: <i>Disruption of external operation wound</i> ... wound infection, and wound breakdown ...
428.0: <i>Congestive heart failure</i> ... DIAGNOSES: 1. Acute congestive heart failure 2. Diabetes mellitus 3. Pulmonary edema ...
202.8: <i>Other malignant lymphomas</i> ... a 55 year-old female with non Hodgkin's lymphoma and acquired C1 esterase inhibitor deficiency ...
770.6: <i>Transitory tachypnea of newborn</i> ... Chest x-ray was consistent with transient tachypnea of the newborn ...
424.1: <i>Aortic valve disorders</i> ... mild aortic stenosis with an aortic valve area of 1.9 cm squared and 2+ aortic insufficiency ...

et al. 2017; Xie and Xing 2018). In terms of data, prior work utilized different domains of data such as radiology reports (Pestian et al. 2007) and death certificates (Koopman et al. 2015), and different modal data such as structured (Perotte et al. 2013) and unstructured text (Scheurwegs et al. 2017). Moreover, some previous work adopted full ICD codes to perform this task (Baumel et al. 2018) while other work adopted partial codes (Xu et al. 2018). Due to such situation, it is difficult to directly compare different work. In this paper, we followed the line of predicting ICD codes from unstructured text of the MIMIC dataset (Johnson et al. 2016), because it is widely studied and publicly available.

The state-of-the-art model for this line of work is the combination of the convolutional neural network (CNN) and the attention mechanism (Mullenbach et al. 2018). However, this model only contains one convolutional layer to build document representations for subsequent layers to predict ICD codes. As shown in Table 1, ICD-related text spans and patterns vary in different examples. Therefore, it may not be sufficient to learn decent document representations from a flat and fixed-length convolutional architecture.

In this paper, we proposed a **Multi-Filter Residual Convolutional Neural Network** (MultiResCNN) for ICD coding using clinical discharge summaries. Our Mul-

tiResCNN model is composed of five layers: the input layer leverages word embeddings pre-trained by word2vec (Mikolov et al. 2013); the multi-filter convolutional layer consists of multiple convolutional filters (Kim 2014); the residual convolutional layer contains multiple residual blocks (He et al. 2016); the attention layer keeps the interpretability for the model following (Mullenbach et al. 2018); the output layer utilizes the sigmoid function to predict the probability of each ICD code.

Our main contribution is that we proposed a novel CNN architecture that combines the multi-filter CNN (Kim 2014) and residual CNN (He et al. 2016). The advantages are two-folds: MultiResCNN not only captures various text patterns with different lengths via the multi-filter CNN, but also enlarges the receptive field¹ (Garcia and Delakis 2004) via the residual CNN. Thus, our model can benefit from rich patterns, the large receptive field and deep architecture. Such method has achieved great success in natural language processing (Vaswani et al. 2017) and computer vision (Krizhevsky, Sutskever, and Hinton 2012).

To evaluate our model, we employed the MIMIC dataset (Johnson et al. 2016) which has been widely used for automated ICD coding. Compared with 5 existing and state-of-the-art models (Perotte et al. 2013; Prakash et al. 2017; Shi et al. 2017; Baumel et al. 2018; Mullenbach et al. 2018), our model outperformed them in nearly all the evaluation metrics (i.e., macro- and micro-AUC, macro- and micro-F1, precision at K). Concretely, in the MIMIC-III experiment using full codes, our model outperformed these models in macro-AUC, micro-F1 and precision at 8 and 15. In the MIMIC-III experiment using top-50 codes and the MIMIC-II experiment using full codes, our model outperformed these models in all evaluation metrics. Moreover, hyper-parameter tuning experiments show that the multi-filter and residual convolutional layers help our model to improve its performance significantly.

Related Work

To the best of our knowledge, the earliest work of automated ICD coding was proposed by Larkey and Croft (1996). They combined three classifiers, K-nearest-neighbor, relevance feedback and Bayesian independence, to assign ICD9 codes to inpatient discharge summaries. However, their method only assigns one code to each discharge summary. Pestian et al. (2007) organized a shared task of assigning ICD-9 codes to radiology reports and their task requires models to assign a large set of codes to each report.

Early work usually used supervised machine learning approaches for ICD coding. Perotte et al. (2013) leveraged “flat” and “hierarchical” Support Vector Machines (SVMs) for automatically assigning ICD9 codes to the discharge summaries of the MIMIC-II repository (Johnson et al. 2016). Their results show that the hierarchical SVM performs better than the flat one. Kavuluru et al. (2015) used the unstructured text in 71,463 EMRs, which come from the University of Kentucky Medical Center, to evaluate supervised learning approaches such as multi-label clas-

sification and learning to rank for the ICD9 code assignment. Koopman et al. (2015) employed the SVM to identify cancer-related causes of death from 447,336 death certificates. Their model is cascaded: the first one identified the presence of cancer and the second identified the type of cancer according to the ICD-10 classification system. Scheurwegs et al. (2017) evaluated coverage-based feature selection methods and Random Forests on seven medical specialties for ICD9 code prediction and two for ICD10, incorporating structured and unstructured text.

With the development of deep learning, researchers also explored neural networks for this task. Shi et al. (2017) utilized the long short-term memory (LSTM) and attention mechanism for automated ICD coding from diagnosis descriptions. Xie and Xing (2018) also adopted the LSTM but they introduced the tree structure and adversarial learning to utilize code descriptions. Prakash et al. (2017) exploited condensed memory neural networks and evaluated it on the free-text medical notes of the MIMIC-III dataset. Baumel et al. (2018) proposed a hierarchical gated recurrent unit (GRU) network, which encodes sentences and documents with two stacked layers, to assign multiple ICD codes to discharge summaries of the MIMIC II and III datasets. Mullenbach et al. (2018) incorporated the convolutional neural network (CNN) with per-label attention mechanism. Their model achieved the state-of-the-art performance among the work using only unstructured text of the MIMIC dataset. Xu et al. (2018) built a hybrid system that includes the CNN, LSTM and decision tree to predict ICD codes from unstructured, semi-structured and structured tabular data. In addition, Lipton et al. (2015) utilized LSTMs to predict diagnostic codes from time series of clinical measurements, while our work focuses on text data.

Method

In this section, we will introduce our **Multi-filter Residual Convolutional Neural Network (MultiResCNN)**, whose architecture is shown in Figure 1. Throughout this paper, we employed the following notation rules: matrices are written as italic uppercase letters (e.g., X); vectors and scalars are written as italic lowercase letters (e.g., x).

Input Layer

Our model leverages a word sequence $w = \{w_1, w_2, \dots, w_n\}$ as input, where n denotes the sequence length. Assuming that \tilde{E} denotes the word embedding matrix, which is pre-trained via word2vec (Mikolov et al. 2013) from the raw text of the dataset. A word w_n will correspond to a vector e_n by looking up \tilde{E} . Therefore, the input will be a matrix $E = \{e_1, e_2, \dots, e_n\} \in \mathbb{R}^{n \times d^e}$.

Multi-Filter Convolutional Layer

To capture the patterns with different lengths, we leveraged the multi-filter convolutional neural network (Kim 2014), where each filter has a different kernel size (i.e., word window size). Assuming we have m filters f_1, f_2, \dots, f_m and their kernel sizes denote as k_1, k_2, \dots, k_m . Therefore, m 1-

¹<http://cs231n.github.io/convolutional-networks/>

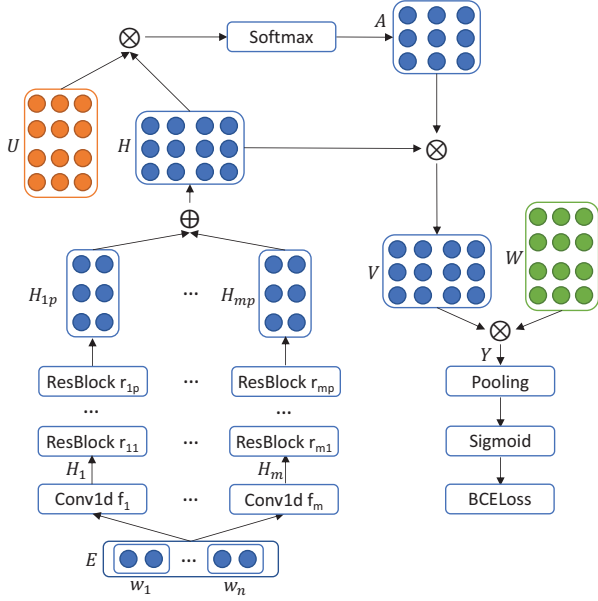


Figure 1: The architecture of our MultiResCNN model. “Conv1d” represents the 1-dimensional convolution, “Res-Block” represents the residual block, “ \oplus ” represents the concatenation operation and “ \otimes ” represents the matrix multiplication. Here we use orange and green for U and W to denote they are learnable parameters, and to distinguish with other matrices (e.g., H) which are not parameters.

dimensional convolutions can be applied to the input matrix E . The convolutional procedure can be formalized as:

$$H_1 = f_1(E) = \bigwedge_{j=1}^n \tanh(W_1^T E^{j:j+k_1-1}), \quad \dots \quad (1)$$

$$H_m = f_m(E) = \bigwedge_{j=1}^n \tanh(W_m^T E^{j:j+k_m-1}),$$

where $\bigwedge_{j=1}^n$ indicates the convolutional operations from left to right. Here we forced the row number n of the output H_1 or $H_m \in \mathbb{R}^{n \times d^f}$ to be the same as that of the input E , because we aimed to keep the sequence length unchanged after convolution. It is simple to implement such goal, e.g., setting the kernel size, padding and stride as k , $\text{floor}(k/2)$ and 1. d^f indicates the out-channel size of a filter and every filter has the same output size.

Moreover, $E^{j:j+k_1-1} \in \mathbb{R}^{k_1 \times d^e}$ and $E^{j:j+k_m-1} \in \mathbb{R}^{k_m \times d^e}$ indicate the sub-matrices of E , starting from the j -th row and ending at the $j+k_1-1$ or $j+k_m-1$ row. $W_1 \in \mathbb{R}^{(k_1 \times d^e) \times d^f}$ and $W_m \in \mathbb{R}^{(k_m \times d^e) \times d^f}$ indicate the weight matrices of corresponding filters. Throughout this paper, the biases of all layers are ignored for conciseness. The overview of a 1-dimensional convolution filter f_m is shown in Figure 2.

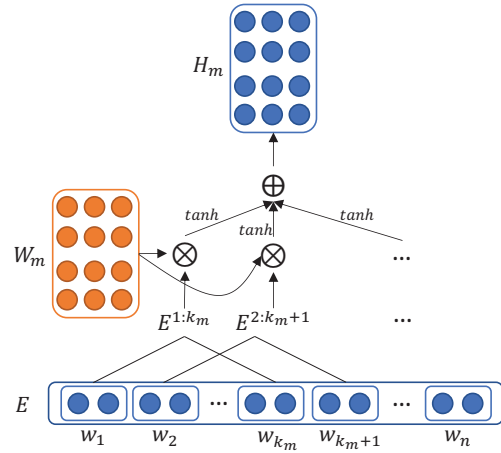


Figure 2: The architecture of a 1-dimensional convolution filter f_m . “ \oplus ” represents the concatenation operation and “ \otimes ” represents the matrix multiplication.

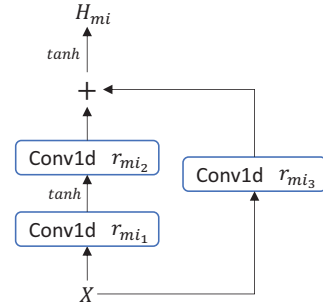


Figure 3: The architecture of a residual block r_{mi} . “+” represents the element-wise addition.

Residual Convolutional Layer

On top of each filter in the multi-filter convolutional layer, there is a residual convolutional layer which consists of p residual blocks (He et al. 2016). Take the m -th filter as an example, the computational procedure of its corresponding residual blocks $r_{m1}, r_{m2}, \dots, r_{mp}$ can be formalized as:

-
- 1: $X = H_m$
 - 2: **for** $i = 1$ **to** p **do**
 - 3: $H_{mi} = r_{mi}(X)$
 - 4: $X = H_{mi}$
 - 5: **return** H_{mp}
-

For the residual block r_{mi} (Figure 3), it consists of three convolutional filters, namely r_{mi1}, r_{mi2} and r_{mi3} . The computational procedure can be denoted as:

$$\begin{aligned}
X_1 &= r_{mi_1}(X) = \bigwedge_{j=1}^n \tanh(W_{mi_1}^T X^{j:j+k_m-1}), \\
X_2 &= r_{mi_2}(X_1) = \bigwedge_{j=1}^n W_{mi_2}^T X_1^{j:j+k_m-1}, \\
X_3 &= r_{mi_3}(X) = \bigwedge_{j=1}^n W_{mi_3}^T X^{j:j}, \\
H_{mi} &= \tanh(X_2 + X_3),
\end{aligned} \tag{2}$$

where $\bigwedge_{j=1}^n$ indicates the convolutional operations. X denotes the input matrix of this residual block and $X^{j:j+k_m-1} \in \mathbb{R}^{k_m \times d^{i-1}}$ indicate the sub-matrices of X , starting from the j -th row and ending at the $j + k_m - 1$ row. $H_{mi} \in \mathbb{R}^{n \times d^i}$ denotes the output matrix of the residual block. d^{i-1} and d^i denote the in-channel and out-channel sizes of this residual block. Therefore, the in-channel size of the first residual block r_{m1} should be d^f and the out-channel size of the last residual block r_{mp} is defined as d^p . Similar with the multi-filter convolutional layer, we let the row numbers of H_{mi} as well as X_1, X_2 and $X_3 \in \mathbb{R}^{n \times d^i}$ be n , which is identical to that of the input X .

Moreover, $W_{mi_1} \in \mathbb{R}^{(k_m \times d^{i-1}) \times d^i}$, $W_{mi_2} \in \mathbb{R}^{(k_m \times d^i) \times d^i}$ and $W_{mi_3} \in \mathbb{R}^{(1 \times d^{i-1}) \times d^i}$ denote the weight matrices of the three convolutional filters, r_{mi_1}, r_{mi_2} and r_{mi_3} . Thereinto, r_{mi_1} and r_{mi_2} have the same kernel size k_m with the corresponding filter f_m in the multi-filter convolutional layer, but they have different in-channel sizes. r_{mi_3} is a special convolutional filter whose kernel size is 1.

Because the m -th filter f_m in the multi-filter convolutional layer corresponds to p residual blocks $r_{m1}, r_{m2}, \dots, r_{mp}$ in the residual convolutional layer, we employed the output $H_{mp} \in \mathbb{R}^{n \times d^p}$ of the p -th residual block r_{mp} as the output of these residual blocks. Since there are totally m filters in the multi-filter convolutional layer, the final output of the residual convolutional layer is a concatenation of the output of m residual blocks, namely $H = H_{1p} \oplus H_{2p} \dots H_{mp} \in \mathbb{R}^{n \times (m \times d^p)}$.

Attention Layer

Following Mullenbach et al. (2018), we employed the per-label attention mechanism to make each ICD code attend to different parts of the document representation H . The attention layer is formalized as:

$$\begin{aligned}
A &= \text{softmax}(HU), \\
V &= A^T H,
\end{aligned} \tag{3}$$

where $U \in \mathbb{R}^{(m \times d^p) \times l}$ represents the parameter matrix of the attention layer, $A \in \mathbb{R}^{n \times l}$ represents the attention weights for each pair of an ICD code and a word, $V \in \mathbb{R}^{l \times (m \times d^p)}$ represents the output of the attention layer. Here l denotes the number of ICD codes.

Output Layer

In the output layer, V is first fed into a linear layer followed by the sum-pooling operation to obtain the score vector \hat{y} for all ICD codes, and then the probability vector \tilde{y} is calculated from \hat{y} by the sigmoid function. This process can be formalized as:

$$\begin{aligned}
Y &= VW, \text{ where } Y \in \mathbb{R}^{l \times l}, \\
\hat{y} &= \text{pooling}(Y), \text{ where } \hat{y}_i = \sum_{j=1}^l Y_{ij}, \\
\tilde{y} &= \text{sigmoid}(\hat{y}),
\end{aligned} \tag{4}$$

where $W \in \mathbb{R}^{(m \times d^p) \times l}$ is the weight matrix of the output layer. For training, we treated the ICD coding task as a multi-label classification problem following previous work (McCallum 1999; Mullenbach et al. 2018). The training objective is to minimize the binary cross entropy loss between the prediction \tilde{y} and the target y :

$$L(w, y, \theta) = - \sum_{j=1}^l y_j \log(\tilde{y}_j) + (1 - y_j) \log(1 - \tilde{y}_j), \tag{5}$$

where w denotes the input word sequence and θ denotes all the parameters. We utilized the back-propagation algorithm and Adam optimizer (Kingma and Ba 2014) to train our model.

Experiments

Datasets

MIMIC-III In this paper, we employed the third version of Medical Information Mart for Intensive Care (MIMIC-III) (Johnson et al. 2016) as the first dataset to evaluate our models. Following Mullenbach et al. (2018), we used discharge summaries, split them by patient IDs, and conducted experiments using the full codes as well as the top-50 most frequent codes. Finally, the MIMIC-III dataset using 8,921 ICD-9 codes consists of 47,719, 1,631 and 3,372 discharge summaries for training, development and testing respectively. The dataset using top-50 codes has 8,067 discharge summaries for training, 1,574 for development, and 1,730 for testing.

MIMIC-II Besides the MIMIC-III dataset, we also leveraged the MIMIC-II dataset to compare our models with the ones in previous work (Perotte et al. 2013; Mullenbach et al. 2018; Baumel et al. 2018). Follow their experimental setting, there are 20,533 and 2,282 clinical notes for training and testing, and 5,031 unique ICD-9 codes in the dataset.

Preprocessing Following previous work (Mullenbach et al. 2018), the text was tokenized, and each token were transformed into its lowercase. The tokens that contain no alphabetic characters were removed such as numbers and punctuations. The maximum length of a token sequence is 2,500 and the one that exceeds this length will be truncated. We

Table 2: Performance comparisons using different configurations in the multi-filter and residual convolutional layers. k denotes the kernel sizes k_1, k_2, \dots, k_m and p denotes the residual block number.

Model	Config	MIMIC-III, full codes			MIMIC-III, top-50 codes		
		P@8	Micro-F1	Macro-F1	P@5	Micro-F1	Macro-F1
CNN	$k=9$	0.706	0.508	0.053	0.590	0.592	0.519
MultiCNN	$k=5,9,15$	0.731	0.534	0.061	0.616	0.633	0.556
	$k=3,5,9,15,19$	0.735	0.542	0.067	0.630	0.646	0.576
	$k=3,5,9,15,19,25$	0.736	0.545	0.068	0.633	0.652	0.584
ResCNN	$p=1$	0.714	0.532	0.063	0.618	0.645	0.560
	$p=2$	0.713	0.532	0.059	0.589	0.601	0.531
	$p=3$	0.710	0.529	0.059	0.575	0.585	0.500
MultiResCNN	$k=3,5,9,15,19,25$ $p=1$	0.741	0.561	0.073	0.638	0.673	0.608

utilized the scripts² provided by Mullenbach et al. (2018) for preprocessing.

Evaluation Metrics

To compare with previous work, we utilized different evaluation metrics in different experiments. In the MIMIC-III experiment using full ICD codes, we utilized macro-averaged and micro-averaged AUC (area under the ROC, i.e., receiver operating characteristic curve), macro-averaged and micro-averaged F1, precision at 8 (P@8) and precision at 15 (P@15). When computing macro-averaged AUC or F1, we first computed the performance for each label and then averaged them. When computing micro-averaged AUC or F1, we considered every pair of a clinical note and a code as an independent prediction. The precision at K (P@K) indicates the proportion of the correctly-predicted labels in the top-K predicted labels.

In the MIMIC-III experiment using the top-50 ICD codes, we employed the P@5 besides macro-averaged and micro-averaged AUC, macro-averaged and micro-averaged F1. In the MIMIC-II experiment using full codes, we employed the same evaluation metrics except that P@5 was changed to P@8.

Hyper-parameter Tuning

Since our model has a number of hyper-parameters, it is infeasible to search optimal values for all hyper-parameters. Therefore, some hyper-parameter values were chosen empirically or following prior work (Mullenbach et al. 2018). The word embedding size d^e is 100, the out-channel size d^f of a filter in the multi-filter convolutional layer is 100, the learning rate is 0.0001, the batch size is 16 and the dropout rate is 0.2.

To explore a better configuration for the filter number m and the kernel sizes k_1, k_2, \dots, k_m in the multi-filter convolutional layer, and the residual block number p in the residual convolutional layer, we conducted the following experiments. First, we developed three variations:

- CNN, which only has one convolutional filter and is equivalent to the CAML model (Mullenbach et al. 2018).

- MultiCNN, which only has the multi-filter convolutional layer.
- ResCNN, which only has the residual convolutional layer.

Then we tried several configurations for these models on the development set of MIMIC-III using the full and top-50 code settings. The experimental results are shown in Table 2. For each configuration, we tried three runs by initializing the model parameters randomly. The results shown in the table are the means of three runs. We selected such kernel sizes since they do not only capture various text patterns from different granularities, but also keeps the sequence length unchanged after convolution (e.g., setting the padding and stride sizes as $\text{floor}(k/2)$ and 1). In addition, we pre-defined the in-channel and out-channel sizes of residual blocks empirically:

- $p=1$: $d^0=100, d^1=50$
- $p=2$: $d^0=100, d^1=100, d^2=50$
- $p=3$: $d^0=100, d^1=150, d^2=100, d^3=50$

As shown in Table 2, MultiCNN performs better than CNN. As the kernel number increases, the performance increases consistently in both full and top-50 code settings. The performance reaches a peak when the kernel sizes are 3,5,9,15,19,25. Moreover, ResCNN also performs better than CNN, but the difference is that the performances deteriorate as the residual block number increases. ResCNN achieves the best performance when the residual block number is 1. Therefore, we applied the best configuration of MultiCNN and ResCNN to MultiResCNN. The results show that the performance of MultiResCNN was further improved after combining MultiCNN and ResCNN. Therefore, we kept such configuration in other experiments.

Baselines

CAML & DR-CAML The Convolutional Attention network for Multi-Label classification (CAML) was proposed by Mullenbach et al. (2018). It has achieved the state-of-the-art results on the MIMIC-III and MIMIC-II datasets among the models using unstructured text. It consists of one convolutional layer and one attention layer to generate label-aware features for multi-label classification (McCallum 1999). The

²<https://github.com/jamesmullenbach/caml-mimic>

Table 3: MIMIC-III results (full codes). The results of MultiResCNN are shown in means \pm standard deviations.

Model	AUC		F1		P@K	
	Macro	Micro	Macro	Micro	8	15
CAML (Mullenbach et al. 2018)	0.895	0.986	0.088	0.539	0.709	0.561
DR-CAML (Mullenbach et al. 2018)	0.897	0.985	0.086	0.529	0.690	0.548
MultiResCNN	0.910	0.986	0.085	0.552	0.734	0.584
	± 0.002	± 0.001	± 0.007	± 0.005	± 0.002	± 0.001

Table 4: MIMIC-III results (top-50 codes). The results of MultiResCNN are shown in means \pm standard deviations.

Model	AUC		F1		P@5
	Macro	Micro	Macro	Micro	
C-MemNN (Prakash et al. 2017)	0.833	-	-	-	0.420
C-LSTM-Att (Shi et al. 2017)	-	0.900	-	0.532	-
CAML (Mullenbach et al. 2018)	0.875	0.909	0.532	0.614	0.609
DR-CAML (Mullenbach et al. 2018)	0.884	0.916	0.576	0.633	0.618
MultiResCNN	0.899	0.928	0.606	0.670	0.641
	± 0.004	± 0.002	± 0.011	± 0.003	± 0.001

Description Regularized CAML (DR-CAML) is an extension of CAML and incorporates the text description of each code to regularize the model.

C-MemNN The Condensed Memory Neural Network was proposed by Prakash et al. (2017), which equips the neural network with iterative condensed memory representations. The model achieved competitive results to predict the top-50 ICD codes for the medical notes in the MIMIC-III dataset.

C-LSTM-Att Shi et al. (2017) proposed a Character-aware LSTM-based Attention model to assign ICD codes to clinical notes. They employed LSTM-based language models to generate representations of clinical notes and ICD codes, and proposed an attention method to address the mismatch between notes and codes. They also focused on predicting the top-50 ICD codes for the medical notes in the MIMIC-III dataset.

SVM Perotte et al. (2013) experimented two approaches: one treats each ICD9 code independently (flat SVM) and the other uses the hierarchical nature of ICD9 codes (hierarchy SVM). Their results show that the hierarchy SVM performs better than the flat one, yielding 29.3% f1-measure in the MIMIC-II dataset.

HA-GRU Baumel et al. (2018) presented a model named Hierarchical Attention Gated Recurrent Unit (HA-GRU) for automatic ICD coding of clinical documents. HA-GRU includes two main layers: the first one encodes sentences and the second one encodes documents. They reported their results in the MIMIC-II dataset, following the data split from Perotte et al. (2013).

Results

In this section, we compared our model with existing work for automated ICD coding. We ran our model three times for each experiment and each time we used different random seeds for parameter initialization. The final results are the

means and standard deviations of three runs. Following prior work (Mullenbach et al. 2018), we compared our model with existing work using the MIMIC-III and MIMIC-II dataset. For the MIMIC-III dataset, we also performed the comparisons with two experimental settings, namely using the full codes and top-50 codes. For the MIMIC-II dataset, only the full codes were employed.

MIMIC-III Results (full codes) As shown in Table 3, we can see that our model obtained better results in the macro-AUC, micro-F1, precision@8 and precision@15, compared with the state-of-the-art models, CAML and DR-CAML. Our model improved the macro-AUC by 0.013, the micro-F1 by 0.013, the precision@8 by 0.025, the precision@15 by 0.023. In addition, our model achieved comparable performance on the micro-AUC and a slightly worse macro-F1. More importantly, we observed that our model is able to attain stable good results from the standard deviations.

MIMIC-III Results (top-50 codes) From Table 4, we observed that our model outperformed all the baselines, namely C-MemNN (Prakash et al. 2017), C-LSTM-Att (Shi et al. 2017), CAML and DR-CAML (Mullenbach et al. 2018), in all evaluation metrics. Our model improves the macro-AUC, micro-AUC, macro-F1, micro-F1 and precision@5 by 0.015, 0.012, 0.030, 0.037 and 0.023, respectively. Our model outperformed the C-MemNN by 0.221 and 0.066 in precision@5 and macro-AUC. It also outperformed the C-LSTM-Att by 0.138 and 0.028 in micro-F1 and micro-AUC. Its precision@5 is 0.032 and 0.023 higher than those of CAML and DR-CAML.

MIMIC-II Results (full codes) Table 5 shows the results on the full code set of MIMIC-II. Perotte et al. (2013) used the SVM to predict ICD codes from clinical text and their method obtained 0.293 micro-F1. By contrast, our model outperformed their method by 0.171 in micro-F1. Baumel et al. (2018) utilized the attention mechanism and GRU for automated ICD coding. Our model outperformed their model by 0.098 in micro-F1. Our model also outperformed

Table 5: MIMIC-II results (full codes). The results of MultiResCNN are shown in means \pm standard deviations.

Model	AUC		F1		P@8
	Macro	Micro	Macro	Micro	
SVM (Perotte et al. 2013)	-	-	-	0.293	-
HA-GRU (Baumel et al. 2018)	-	-	-	0.366	-
CAML (Mullenbach et al. 2018)	0.820	0.966	0.048	0.442	0.523
DR-CAML (Mullenbach et al. 2018)	0.826	0.966	0.049	0.457	0.515
MultiResCNN	0.850	0.968	0.052	0.464	0.544
	± 0.002	± 0.001	± 0.002	± 0.002	± 0.007

Table 6: Analysis of the computational cost between CAML and MultiResCNN. “m”, “s”, “ep” and “d” denote million, second, epoch and document respectively.

	CAML	MultiResCNN
Parameter Amount	6.2m	11.9m
Training Time	438s/ep	1026s/ep
Training Epoch	85	26
Inference Speed	108.7d/s	70.9d/s

the state-of-the-art model, CAML or DR-CAML, by 0.024, 0.002, 0.003, 0.007 and 0.021 in all evaluation metrics.

Discussion

Computational Cost Analysis

In this section, we analyzed the computational cost between the state-of-the-art model, CAML and our model, MultiResCNN. The analysis was conducted from four aspects, namely the parameter amount, training time, training epoch, inference speed. Our experimental settings are as follows. For CAML, we used the optimal hyper-parameter setting reported in their paper (Mullenbach et al. 2018). For MultiResCNN, we used six filters and 1 residual block, which obtained the best result in our hyper-parameter tuning experiments. The batch size, learning rate and dropout rate are identical in every experiment. We used the training set and development set of MIMIC-III (full codes) as experimental data. The experiments were conducted on NVIDIA Tesla P40 GPUs. Training will terminate if the performance on the development set does not increase for 10 times.

As shown in Table 6, the parameter of MultiResCNN is approximately 1.9 times as many as that of CAML. The training time of MultiResCNN is about 2.3 times more than that of CAML. It is reasonable since MultiResCNN has more filters and layers. Interestingly, MultiResCNN needs much less epochs to converge. Considering the inference speed, CAML is approximately 1.5 times faster than MultiResCNN. Overall, the computational cost of MultiResCNN is larger than that of CAML, but we hold the opinion that the increased cost is still acceptable.

Effect of Truncating Data

During preprocessing, we truncated the discharge summaries that are longer than 2,500 tokens. To investigate the effect of the length limitation, we further conducted the

experiments using 3,500, 4,500, 5,500 and 6,500. We selected these values because the maximum length of the discharge summaries in the development set is approximately 6,300. Results show that the performance differences between different settings are not significant. P@8 ranges between 0.736 and 0.741, and micro-F1 ranges between 0.557 and 0.566. 2,500 seems to be a decent selection considering the tradeoff between performance and cost.

Limitations

In this study, the performance improvement mostly comes from deep and diversified representations of text. In the future, we will explore how to incorporate BERT (Devlin et al. 2019) into this task effectively and efficiently. In our preliminary experiments, BERT did not perform well due to the limitations of hardware and its fixed-length context. Therefore, potential solutions include recurrent Transformer (Dai et al. 2019) and hierarchical BERT (Zhang, Wei, and Zhou 2019). Moreover, we chose the kernel sizes of the multi-filter layer and channel sizes of the residual layer empirically, which should be further studied and optimized in the future.

Conclusions

In this paper, we proposed a multi-filter residual convolutional neural network for ICD coding. We conducted three experiments on the widely-used MIMIC-III and MIMIC-II datasets. Results show that our model achieved the state-of-the-art performance compared with several competitive baselines. We found that both multi-filter convolution and residual convolution helped the performance improvement with acceptable computational cost. This shows deep and diversified text representations could benefit the ICD coding from clinical text. Our model can be a strong baseline for not only ICD coding, but also other text classification tasks.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, R01DA045816, R01HL125089, R01HL137794, R01HL135219, and R01LM012817. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Baumel, T.; Nassour-Kassis, J.; Cohen, R.; Elhadad, M.; and 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*.
- Bottle, A., and Aylin, P. 2008. Intelligent information: a national system for monitoring clinical performance. *Health services research* 43(1p1):10–31.
- Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the ACL*, 2978–2988.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.
- Garcia, C., and Delakis, M. 2004. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence* 26(11):1408–1423.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3:160035.
- Kavuluru, R.; Rios, A.; and Lu, Y. 2015. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine* 65(2):155–166.
- Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Koopman, B.; Zuccon, G.; Nguyen, A.; Bergheim, A.; and Grayson, N. 2015. Automatic icd-10 classification of cancers from free-text death certificates. *International journal of medical informatics* 84(11):956–965.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems* 25. Curran Associates, Inc. 1097–1105.
- Larkey, L. S., and Croft, W. B. 1996. Combining classifiers in text categorization. In *SIGIR*, volume 96, 289–297. Citeseer.
- Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. 2015. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*.
- McCallum, A. 1999. Multi-label text classification with a mixture model trained by em. In *AAAI workshop on Text Learning*, 1–7.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Mullenbach, J.; Wiegrefe, S.; Duke, J.; Sun, J.; and Eisenstein, J. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1101–1111.
- O’Malley, K. J.; Cook, K. F.; Price, M. D.; Wildes, K. R.; Hurdle, J. F.; and Ashton, C. M. 2005. Measuring diagnoses: Icd code accuracy. *Health services research* 40(5p2):1620–1639.
- Perotte, A.; Pivovarov, R.; Natarajan, K.; Weiskopf, N.; Wood, F.; and Elhadad, N. 2013. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association* 21(2):231–237.
- Pestian, J. P.; Brew, C.; Matykiewicz, P.; Hovermale, D. J.; Johnson, N.; Cohen, K. B.; and Duch, W. 2007. A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, 97–104. Association for Computational Linguistics.
- Prakash, A.; Zhao, S.; Hasan, S. A.; Datla, V.; Lee, K.; Qadir, A.; Liu, J.; and Farri, O. 2017. Condensed memory networks for clinical diagnostic inferencing. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Scheurwegs, E.; Cule, B.; Luyckx, K.; Luyten, L.; and Daelemans, W. 2017. Selecting relevant features from the electronic health record for clinical code prediction. *Journal of biomedical informatics* 74:92–103.
- Shi, H.; Xie, P.; Hu, Z.; Zhang, M.; and Xing, E. P. 2017. Towards automated icd coding using deep learning. *arXiv preprint arXiv:1711.04075*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Xie, P., and Xing, E. 2018. A neural architecture for automated icd coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1066–1076.
- Xu, K.; Lam, M.; Pang, J.; Gao, X.; Band, C.; Xie, P.; and Xing, E. 2018. Multimodal machine learning for automated icd coding. *arXiv preprint arXiv:1810.13348*.
- Zhang, X.; Wei, F.; and Zhou, M. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the ACL*, 5059–5069.