Medical Code Assignment with Gated Convolution and Note-Code Interaction

Shaoxiong Ji †, **Shirui Pan** ‡, **Pekka Marttinen** †
† Department of Computer Science, Aalto University
‡ Faculty of Information Technology, Monash University
{shaoxiong.ji; pekka.marttinen}@aalto.fi
shirui.pan@monash.edu

Abstract

Medical code assignment from clinical text is a fundamental task in clinical information system management. As medical notes are typically lengthy and the medical coding system's code space is large, this task is a long-standing challenge. Recent work applies deep neural network models to encode the medical notes and assign medical codes to clinical documents. However, these methods are still ineffective as they do not fully encode and capture the lengthy and rich semantic information of medical notes nor explicitly exploit the interactions between the notes and codes. We propose a novel method, gated convolutional neural networks, and a note-code interaction (GatedCNN-NCI), for automatic medical code assignment to overcome these challenges. Our methods capture the rich semantic information of the lengthy clinical text for better representation by utilizing embedding injection and gated information propagation in the medical note encoding module. With a novel note-code interaction design and a graph message passing mechanism, we explicitly capture the underlying dependency between notes and codes, enabling effective code prediction. A weight sharing scheme is further designed to decrease the number of trainable parameters. Empirical experiments on real-world clinical datasets show that our proposed model outperforms state-of-the-art models in most cases, and our model size is on par with light-weighted baselines.

1 Introduction

Automatic medical code assignment is a routine healthcare task for medical information management and clinical decision support. The International Classification of Diseases (ICD) coding system, maintained by the World Health Organization (WHO), is widely used among various coding systems. Thus, the medical code assignment task is also called ICD coding. It uses all types of clinical notes to predict medical codes in a supervised manner with human-annotated codes (Perotte et al., 2014), which is formulated as a multi-class multi-label text classification problem in the medical domain.

While there are increasing works in the community in automatic medical code assignment (Prakash et al., 2017; Shi et al., 2017; Mullenbach et al., 2018; Ji et al., 2020), this task remains challenging from the perspectives of note representation and code prediction. First, medical note representation, a critical step in understanding medical notes, is formidably challenging due to the lengthy and complex semantic information in the discharge documents. There are typically thousands of tokens in a medical note due to the various diagnoses and procedures experienced by a patient. Furthermore, clinical notes also contain a vocabulary with many professional words and phrases, making it hard for a neural network model to encode and understand critical information. Second, the medical coding system has a very high and sparse dimensional label space, which renders the code prediction task incredibly difficult. For example, ICD9 and ICD10 coding systems have many labels, i.e., more than 14,000 and 68,000 codes. However, a patient typically is diagnosed with only a couple of codes over the whole coding space.

Early works for medical code assignment typically follow statistical approaches. They either employ rule-based methods (Farkas and Szarvas, 2008) or apply classification methods such as SVM and Bayesian ridge regression (Lita et al., 2008) to assign the codes. These methods are shallow and do not exploit the complex semantic information in medical notes, leading to unsatisfactory performance. Recently, Natural language processing (NLP) techniques based on deep learning have been developed (Mullenbach et al., 2018; Li et al.,

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1034–1043
August 1–6, 2021. ©2021 Association for Computational Linguistics

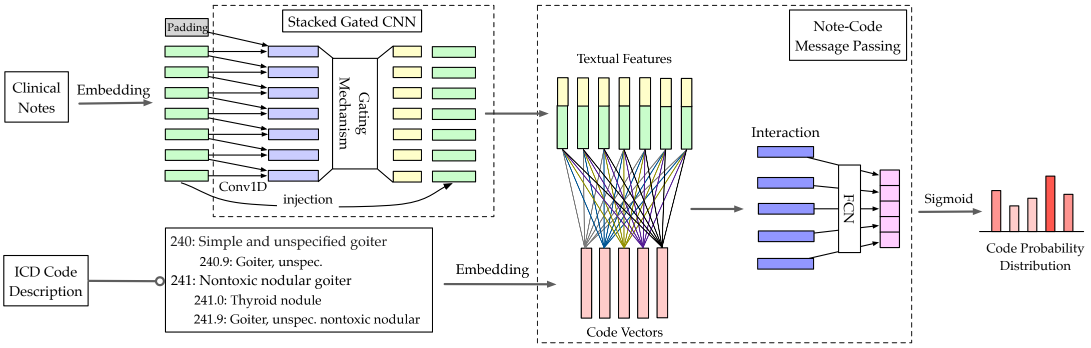

Figure 1: Illustration of the GatedCNN-NCI model architecture. The gating mechanism controls the information propagation. Textual features interact with each code vector in the note-code interaction module. FCN is a fully connected layer.

2018; Cao et al., 2020; Ji et al., 2020), which learn the note representation via convolutional neural networks. Specifically, CAML (Mullenbach et al., 2018), MultiResCNN (Li et al., 2018) and DCAN (Ji et al., 2020) treat ICD coding as a general text classification problem and develop complex neural encoders to learn the note representation. HyperCore (Cao et al., 2020) proposes the hyperbolic embedding to capture code hierarchy and co-occurrence. However, these approaches are still ineffective, as they do not explicitly capture the fine-grained interactions between textual elements and medical codes. These interactions naturally represent the interdependencies between the complex medical words and associated codes, and thus should be well exploited.

This paper puts forward a novel neural architecture, Gated Convolutional Neural Network with Note-Code Interaction (GatedCNN-NCI), for effective medical code assignment. Our goal is to learn rich representation from clinical notes and exploit the interactions between medical texts and clinical codes. To capture the long sequential history of clinical documents, we design a novel dilation information propagation component with a forgetting mechanism to selectively utilize the useful information for note representation learning. To tackle the large labeling space, we formulate textual notes and medical codes as a complete bipartite graph and develop a graph message passing approach to capture the explicit interaction between notes and codes. The ICD code descriptions are used as an external medical knowledge source to learn more accurate code representations that preserve the semantic relations of the codes. Considering the practical application in real-world medical institutes, especially those with limited computing resources, our architecture also prioritizes computational efficiency when designing the sub-modules. Our contributions are itemized as follows.

- We propose a CNN-based neural architecture with dilation and gating mechanism for clinical text encoding. We enhance the feature representation learning with 1) embedding injection, enhancing the deeper features of lengthy clinical notes; 2) and the gating mechanism to control the information propagation.
- We view the note-code interaction as a complete bipartite graph and propose a graph message passing mechanism to capture the interactions between textual features and ICD codes explicitly.
- To reduce the trainable parameters and make our model computationally efficient, we develop a weight-sharing mechanism across the length of the sequence and the depth of the network.
- Experiments in real-world clinical datasets empirically validate our model's effectiveness by comparison with the state of the art.

2 Related Work

Classical medical coding systems used rule-based methods (Farkas and Szarvas, 2008), studied feature selection (Medori and Fairon, 2010), and applied classification models such as SVM and Bayesian ridge regression (Lita et al., 2008). Perotte et al. (2014) utilized the hierarchical structure of the ICD code systems and provided a flat and hierarchical SVM for diagnosis code classification, while Kavuluru et al. (2015) studied explicit co-occurrence relations between codes. Scheurwegs et al. (2016) investigated heterogeneous data of both structured records and textual data. Recent deep learning-based models use word embedding techniques and develop complex neural network architectures to learn rich text features for automatic medical code assignment. Popular models use recurrent architectures such as the LSTM network with an attention mechanism (Shi et al., 2017) and GRU network with hierarchical attention (Baumel et al., 2018). Prakash et al. (2017) used Wikipedia as a knowledge source and proposed condensed memory networks (C-MemNNs) with iterative condensation of memory representation. Although CNNs are traditionally applied in computer vision, many ICD coding methods utilize convolutional architectures. CAML (Mullenbach et al., 2018) used CNN with multiple filters and label attention. Li et al. (2018) adopted the doc2vec embedding and CNN architecture, and Bai and Vucetic (2019) incorporated online knowledge sources. The recent MultiResCNN model (Li and Yu, 2020) extensively concatenated and stacked CNNs with multi-filter convolution and residual learning. HyperCore (Cao et al., 2020) utilized hyperbolic embedding and co-graph representation to capture the code hierarchy.

3 Method

3.1 High-level Model Architecture

3.2 Problem Definition

The input clinical note with n words is denoted as $\mathbf{x}_{1:n} = x_1, \ldots, x_n$, where each x_i is a word (or token). The medical coding system is the set of all possible diagnosis and procedure codes denoted as $\mathcal{C}$. The medical code assignment learns a function $\mathcal{F} : \mathcal{X}^n \to \mathcal{Y}^m$ such that

$$y = \mathcal{F}\left(x_1, \ldots, x_n; \mathcal{D}\right), \tag{1}$$

where $y \in \mathbb{R}^m$ is the medical code at discharge, m is the number of medical codes, and $\mathcal{D}$ is an optional external knowledge source. This paper uses the ICD coding system and naturally utilizes the official textual ICD code description as an external knowledge source.

The high-level model architecture of GatedCNN-NCI is illustrated in Fig. 1. Our model consists of two main components, i.e., stacked gated CNN layers for clinical note encoding and note-code interaction to fuse the external ICD code description. The stacked gated CNNs include three sub-modules, i.e., dilated convolution, embedding injection, and gating mechanism.

We use word2vec (Mikolov et al., 2013) to train word embeddings from raw tokens. Word embedding matrix of a clinical note is denoted as $[\mathbf{w}_1, \ldots, \mathbf{w}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times d_e}$, where d_e is the dimension of word vectors. Then we input the word embeddings into stacked gated CNN layers for long-range information propagation. The stacked module uses dilated convolution as its backbone (Oord et al., 2016). To further enhance the feature learning, we inject the original embedding into each stacked layer. The gating mechanism is originated from the long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997). We adopt the LSTM-like gate (Dauphin et al., 2017) to control the information flow.

To avoid blurry memory in higher layers, we inject the original word embeddings (Bai et al., 2019). Label interaction has been studied by Wang and Jiang (2016) and Du et al. (2019). We utilize descriptive knowledge from the ICD code descriptions and develop the note-code interaction to capture the relational match between clinical note features and ICD codes. To reduce the training cost and stabilize the training process, we also introduce a weight sharing mechanism across the stacked CNNs (Bai et al., 2019).

3.3 Dilated Convolutional Layers

We use the one-dimensional convolution with dilation as the backbone of our encoder, which takes the word embedding $\mathbf{X} \in \mathbb{R}^{n \times d_e}$ as input. Dilated CNN has exhibited a significant capacity for long sequence modeling and computationally efficient for parallelism (Bai et al., 2018). Specifically, we use a 1D convolution operator $\mathrm{Conv1D}(x; f)$, with a filter $f : \{0, \ldots, k-1\} \to \mathbb{R}$, to each dimension of the word vectors. Given a sequence of one-dimensional elements $\mathbf{x} \in \mathbb{R}^n$, the one-dimensional dilated convolution $\mathcal{F}_d$ is denoted as

$$\mathcal{F}_d(s) = \left(\mathbf{x} *_d f\right)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d\cdot i}, \tag{2}$$

where d is the dilation size (i.e., the space between kernel elements), s is the index of the element of the input sequence, k is the convolving kernel (aka, the filter) size, and $s - d \cdot i$ refers to past time steps. The dilation size of d and kernel size k control the receptive field. The 1D dilated convolution has d_h output channels, i.e., for each of the d_e input channels d_h convolutional features are learned through the dilated $\mathrm{Conv1D}$. Stacking CNN layers can be adopted to learn in-depth features.

3.4 Embedding Injection

Our hypothesis for encoding a very long clinical sequence is that the deep neural encoding architecture tends to forget important information, mainly because the clinical note contains fruitful professional expression about the patient's diagnosis. Thus, in-depth features become blurry with the increase of neural layers. We propose to inject original word embedding into each intermediate layer of the proposed architecture, attempting to remind the network to reactivate the original diagnostic notes and mitigate the failure of extracting meaningful, in-depth features. We denote the hidden representation at the l-th layer as $\mathbf{H}^l \in \mathbb{R}^{n \times d_h}$, where the dimension d_h is the hidden dimension. Word embedding is concatenated into lth-layer hidden representation as

$$\mathbf{J}^l = \mathrm{concat}[\mathbf{X}, \mathbf{H}^l], \tag{3}$$

where $\mathbf{J}^l \in \mathbb{R}^{n \times (d_e + d_h)}$ are the deep features enhanced with the original clues, used as the new input of the next convolutional encoding layer. We randomly initialize the $\mathbf{H}^0$ matrix for the first convolutional layer.

3.5 Gating Mechanism

Embedding injection of original word vectors brings low-level features to higher-level, which may lead to difficulty in feature learning in higher layers. Thus, we develop an LSTM-style gating mechanism to control the information flow and capture a long history in the sequence. Unlike the recurrent gate such as the LSTM that controls the information flow along the time coordinate, this gating mechanism controls the flow through stacked layers' depth. The gating mechanism is depicted in Fig. 2, where σ and $\tanh$ are sigmoid and hyperbolic tangent activation functions respectively. After the embedding injection, the dilated CNN upsamples the injected signal $\mathbf{J}^l$ into $\mathbf{U}^l \in \mathbb{R}^{n \times d_u}$ at the l-th layer. We divide $\mathbf{U}^l$ into four matrices with the same dimension, i.e., $\mathbf{I}$, $\mathbf{O}$, $\mathbf{G}$ and $\mathbf{F} \in \mathbb{R}^{n \times d_g}$, such that:

$$\mathbf{U}^l = \mathrm{concat}[\mathbf{I}, \mathbf{O}, \mathbf{G}, \mathbf{F}]. \tag{4}$$

Here, we have $d_u = 4 \times d_g$. Then, these four matrices are fed into the LSTM-like gating module that controls what information should be propagated to deeper layers. The input gate $\sigma(\mathbf{I})$ decides the information to be infused and stored into the cell state $\mathbf{C}$. The forget gate $\sigma(\mathbf{F})$ chooses the information to be remembered. The output gate $\sigma(\mathbf{O})$, working with the cell state, focuses on what signals propagate into the next layer. This process is formalized as

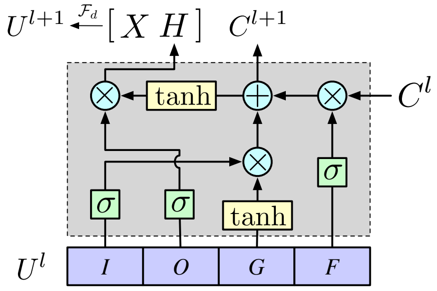

Figure 2: Gating mechanism that controls the flow's convolutional features through stacked layers' depth.

$$\mathbf{C}^{l+1} = \sigma(\mathbf{F}) * \sigma(\mathbf{C}^l) + \sigma(\mathbf{I}) * tanh(\mathbf{G})$$
$$\mathbf{H} = \sigma(\mathbf{O}) * tanh(\mathbf{F}),$$

where $\mathbf{C}^l$ is the cell state at the l-th layer and $\mathbf{H}$ is the hidden state produced by the gated unit. The embedding injection trick concatenates the original word embedding $\mathbf{X}$ and the hidden representation $\mathbf{H}$, and the dilated convolutional layer upsamples the concatenation to get the new feature $\mathbf{U}^{l+1}$ at the $(l+1)$-th layer, denoted as:

$$\mathbf{U}^{l+1} \xleftarrow{\mathcal{F}_d} [\mathbf{X}, \mathbf{H}]. \tag{5}$$

Gated CNNs can be stacked into a deep architecture, as shown in the general framework of Fig. 1. As a result, our model can represent a large-sized context and extract hierarchical features at each layer. Moreover, the gating mechanism can also extract important features to remember and focus, while less critical features are forgotten and ignored at each layer.

3.6 Note-Code Interaction as Message Passing

To capture the explicit note-code interaction (NCI) between the medical codes and textual mentions, we build a complete bipartite graph $G = \{U, V, E\}$, where $U = \{w_i\}^n$ and $V = \{c_j\}^m$ represent the words and ICD codes respectively, and E is the fully connected edge set. For simplicity, we omit the superscript of the last convolutional features $\mathbf{U}^{l+1}$ extracted by the stacked gated CNNs and denote the textual node features $\mathbf{U}$ as the vertex set U in the note-code bipartite graph. We incorporate the ICD code descriptions of WHO to represent the medical knowledge about ICD codes. For example, the ICD code 240 in Fig. 1 is about simple and unspecified goiter. Instead of merely using the ICD code index to represent the prediction target, we include the code description, which contains rich domain knowledge. Word embeddings of description are averaged to obtain code vectors $\mathbf{V} \in \mathbb{R}^{m \times d_v}$, where m is the number of codes, and d_v is the embedding dimension. We take the code vectors as the node features of the vertex set V.

Our novel formulation of the bipartite graph preserves the source-target matching between textual features and ICD code vectors. We utilize the graph message passing mechanism (Gilmer et al., 2017; Wu et al., 2020) to infer fine-grained clues about dependencies between textual features and code semantics. The composition function $\mathrm{NCI} : \mathbb{R}^{n \times d_u} \times \mathbb{R}^{m \times d_v} \rightarrow \mathbb{R}^m$ is denoted as:

$$\mathrm{NCI}(U, V) = f_\theta\left(\sum_{i,j} g_\xi(w_i, c_j)\right), \qquad (6)$$

where g_ξ with parameter ξ is a neural message function and f_θ with parameter θ is an output function. It takes the textual features of all tokens in a note and embeddings of code vectors as inputs and produces an interaction score between the note and each code. To improve the computational efficiency, we take the dot product as the message function g_ξ. The explicit interaction score between token w_i and code c_j is calculated as

$$\mathbf{I}_{ij} = \mathbf{V}_{i,:}\mathbf{U}^{\mathrm{T}}_{j,:}, \qquad (7)$$

where $\mathbf{V}_{i,:}$ is the row vector of textual features representing the i-th word, $\mathbf{U}_{j,:}$ is the row vector of ICD code matrix representing the j-th code in ICD code set. We set $d_u = d_v$ and get the interaction matrix $\mathbf{I} \in \mathbb{R}^{m \times n}$ with dot product. We use a fully connected network f_θ to calculate the scores of the note-code interactions as output. Similar to the matrix factorization formulation of language models (Yang et al., 2017; Li et al., 2020), this dot-product interaction between notes and codes approximates the point-wise mutual information of note-code co-occurrence.

3.7 Parameter-efficient Weight Sharing

The embedding injection and convolutional feature concatenation make the hidden feature high-dimensional. Moreover, as a result of stacking deep layers, the overall model will become cumbersome. Thus, we utilize a weight sharing mechanism (Bai et al., 2019) to decrease the number of parameters. Specifically, we share the weights of gated CNN layers across time steps and depth through neural layers. This mechanism has two benefits. First, it can decrease the number of trainable parameters because weights across the network are tied. Second, it provides a form of regularization to stabilize the training process.

3.8 Objective and Training

We formulate the ICD code assignment as a multi-label multi-class classification problem. We adopt the binary cross entropy loss denoted as:

$$\mathcal{L} = \sum_{i=1}^{m} \left[-y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i)\right], \qquad (8)$$

where $y_i \in \{0, 1\}$ is the ground-truth label, $\hat{y}_i$ is the sigmoid score for prediction, and m is the number of ICD codes. We use Adam optimizer (Kingma and Ba, 2014) to train the model with backpropagation.

4 Experiments

In the experimental analysis on real-world datasets, we compare our proposed model with several recent strong baselines. Our code is available[1].

4.1 Datasets

This paper focuses on textual discharge summaries from a hospital stay. Specifically, we use raw notes, ICD diagnoses, and procedures for patients from two public clinical datasets, i.e., MIMIC-II and MIMIC-III[2], for experiments. Discharge summaries labeled with a set of ICD-9 diagnosis and procedure codes include descriptions of procedures performed by the physician, diagnosis notes, patient's medical history, and discharge instructions.

MIMIC-II. The first dataset of clinical notes is from the Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II) database (Saeed et al., 2011). We follow the standard train-test split performed by Perotte et al. (Perotte et al., 2014), where 90% and 10% of 22,815 non-empty discharge summaries are used for training and testing, respectively.

MIMIC-III. The second dataset is an updated database from Medical Information Mart for In-

[1] https://agit.ai/jsx/GatedCNN-NCI

[2] https://mimic.physionet.org

tensive Care III (MIMIC-III) repository (Johnson et al., 2016), containing patient admitted to Intensive Care Unit (ICU) at a US medical center during 2001 to 2012. We use the "noteevents" table in the latest version 1.4, with 58,576 hospital admissions. Free-text discharge summaries in the MIMIC-III database are extracted to form the dataset with clinical text. The experimental evaluation considers two settings. The first one uses the full set of ICD codes. Following Shi et al. (Shi et al., 2017) and Mullenbach et al. (Mullenbach et al., 2018), an additional experiment on the subset of MIMIC-III with the top 50 frequent labels is conducted. This MIMIC-III top-50 subset has a train/dev/test split with 8,066, 1,573, and 1,729 samples.

4.2 Settings

Preprocessing We preprocess the textual documents following the preprocessing procedures developed by Mullenbach et al. (2018) and Li and Yu (2020). The NLTK package[3] is utilized for tokenization, and all tokens are converted into lowercase. All words appearing in less than three training documents were replaced with "unk". We truncate all documents at the length of 2500 tokens. The word embeddings are initialized with embedding vectors pre-trained on all discharge notes with the continuous-bag-of-words (CBOW) method of word2vec (Mikolov et al., 2013).

Hyper-parameters Some standard settings follow the prior works. For example, the word embedding dimension is 100, and the dropout rate is 0.2. Adam optimizer (Kingma and Ba, 2014) is used to optimize our model parameters. For the rest hyper-parameters, the random search is utilized to search the optimal settings. The searching range or choices of specific hyper-parameters are listed in Table 1. The searching interval of learning rate is $[1e^{-6}, 1e^{-2}]$. Besides, we optimize for kernel size, levels of residual connections, and hidden representation dimension.

Table 1: Range and choices of hyper-parameter search

Hyper-parameters	Range/choices
Learning rate	$[1e^{-6}, 1e^{-2}]$
Kernel size	2, 3, 5, 9
CNN levels	1, 2, 3, 4, 5
Hidden dimension	100, 200, 300, 400, 500, 600

Evaluation Metrics We use area under the receiver operating characteristic curve (AUC-ROC), F1-score, and precision at k (P@k) for evaluation. We set $k = 5$ for MIMIC-III subset with top-50 frequent codes and $k = 8$ for full sets of MIMIC-II and MIMIC-III. In the multi-label classification setting, we use two averaging strategies, i.e., micro and macro. The macro scores are obtained by averaging the respective label-wise scores across all labels. Micro scores give more weight to frequent labels by considering all labels jointly. We run the experiments for 5 times and report the mean ± standard deviation.

4.3 Baselines

We consider the following baseline models. MultiResCNN (Li and Yu, 2020) and HyperCore (Cao et al., 2020) are two recent strong models with the state-of-the-art performance. **Bi-GRU (Mullenbach et al., 2018)** uses a simplified gated recurrent unit with bi-direction, where last hidden representations are used for classification. **C-MemNN (Prakash et al., 2017)** introduces an iterative condensation of memory representations and utilizes external knowledge source from Wikipedia to enhance memory networks by preserving the hierarchical structure in the memory. **AttentiveLSTM (Shi et al., 2017)** encodes clinical descriptions and ICD long titles jointly with character- and word-level LSTM networks and uses attention mechanism for matching important diagnosis snippets. **CAML (Mullenbach et al., 2018)** integrates CNNs and a label-wise attention mechanism to learn rich representations. It has a variant called DR-CAML that uses ICD code descriptions to regularized the loss function. **LEAM (Wang et al., 2018)** encodes two channels of inputs and leverages the compatibility between word and label embeddings to calculate attention scores. **MultiResCNN (Li and Yu, 2020)** combines residual learning (He et al., 2016) and multiple channels concatenation with different convolutional filters, achieving good performance in most settings. **HyperCore (Cao et al., 2020)** utilizes hyperbolic embedding and co-graph representation with code hierarchy. It gains slightly better performance than the MultiResCNN.

4.4 Results

Our model performs consistently the best for frequent labels. First, it beats all models in the MIMIC-III subset with top-50 codes (columns 2-6 in Table 2). For the micro scores that give more

[3] http://www.nltk.org

Table 2: Results on MIMIC-III with top-50 and full codes. "-" indicates no results reported in the original paper. **Bold** text denotes the best and *italic* text denotes the second best.

Model	MIMIC-III Top-50 Codes					MIMIC-III Full Codes				
	AUC-ROC		F1		P@5	AUC-ROC		F1		P@8
	Macro	Micro	Macro	Micro		Macro	Micro	Macro	Micro	
Bi-GRU (Mullenbach et al., 2018)	82.8	86.8	48.4	54.9	59.1	82.2	97.1	3.8	41.7	58.5
C-MemNN (Prakash et al., 2017)	83.3	-	-	-	42.0	-	-	-	-	-
CNN (Kim, 2014)	87.6	90.7	57.6	62.5	62.0	80.6	96.9	4.2	41.9	58.1
Attentive LSTM (Shi et al., 2017)	-	90.0	-	53.2	-	-	-	-	-	-
DR-CAML (Mullenbach et al., 2018)	88.4	91.6	57.6	63.3	61.8	89.7	98.5	8.6	52.9	69.0
LEAM (Wang et al., 2018)	88.1	91.2	54.0	61.9	61.2	-	-	-	-	-
MultiResCNN (Li and Yu, 2020)	89.9±0.4	92.8±0.2	60.6±1.1	67.0±0.3	64.1±0.1	91.0±0.2	98.6±0.1	8.5±0.7	*55.2*±0.5	*73.4*±0.2
HyperCore (Cao et al., 2020)	89.5±0.3	92.9±0.2	60.9±0.1	66.3±0.1	63.2±0.2	**93.0**±0.1	**98.9**±0.5	*9.0*±0.3	55.1±0.1	72.2±0.2
GatedCNN-NCI (ours)	**91.5**±0.3	**93.8**±0.1	**62.9**±0.5	**68.6**±0.1	**65.3**±0.1	*92.2*±0.2	**98.9**±0.3	**9.2**±0.2	**56.3**±0.1	**73.6**±0.3

weight to frequent labels, our model also has the best predictive metrics (columns 8&10 in Table 2 and columns 3&5 in Table 3). Moreover, our model is competitive also with the rest of the metrics: it consistently has the best P@k scores and at worst, the second best macro scores in all datasets.

MIMIC-III (Top-50 Codes) The first experiment uses the MIMIC-III subset with top-50 codes, showing models' performance on predicting the frequent diagnosis. The results in Table 2 show that our model outperforms all the baselines in all the evaluation metrics. Significantly, our model gains a higher macro F1-score by 2% and micro F1-score by 1.6% than the state of the art.

MIMIC-III (Full Codes) We then run our model on the MIMIC-III dataset with full codes. Our model outperforms most baselines, gaining the best scores in macro AUC-ROC, macro F1, micro F1, and precision@8. For the macro AUC-ROC, our model is ranked at the second place.

MIMIC-II (Full Codes) In the third dataset of MIMIC-II, we also predict the full set of ICD-9 codes. Our model achieves predictive performance on par with two recent strong baselines of MultiResCNN and HyperCore. We gain the best scores in micro AUC-ROC, micro F1-score, and P@8. Macro AUC-ROC and F1 scores of our model are the second best of the models compared.

Table 3: Results on MIMIC-II full codes. **Bold** text denotes the best and *italic* text denotes the second best.

Model	AUC-ROC		F1		P@8
	Macro	Micro	Macro	Micro	
CNN	74.2	94.1	3.0	33.2	38.8
Bi-GRU	78.0	95.4	2.4	35.9	42.0
DR-CAML	82.6	96.6	4.9	45.7	51.5
MultiResCNN	85.0±0.2	96.8±0.1	5.2±0.2	46.4±0.2	*54.4*±0.7
HyperCore	**88.5**±0.1	*97.1*±0.4	**7.0**±0.2	*47.0*±0.3	53.7±0.3
GatedCNN-NCI	*87.2*±0.3	**97.2**±0.1	*6.4*±0.3	**47.3**±0.2	**54.5**±0.4

4.5 Comparison with BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) has revolutionized the NLP community recently. The pre-trained language model has been applied to different downstream NLP tasks. We compare our model's performance with the BERT model and a domain-specific variant, i.e., ClinicalBERT (Alsentzer et al., 2019) pre-trained on the clinical text of MIMIC-III. For the BERT model, we use the uncased BERT-base with a hidden dimension of 768. Because these two BERT models require the configuration of the maximum sequence length of 512, we truncate the text sequence for our model to ensure a fair comparison. BERT models have two special tokens, i.e., `[CLS]` and `[SEP]`. Thus, we truncate clinical notes with a length of 510. We use Huggingface's transformer framework[4] when implementing these two models. The results in Table 4 show that pretraining the language model with domain data improves the performance, and our model has better performance in most evaluation metrics.

Table 4: Comparison with BERT and ClinicalBERT using the MIMIC-III top-50 code dataset with sequence length truncated at 510.

Model	AUC-ROC		F1		P@5
	Macro	Micro	Macro	Micro	
BERT-base	80.6	85.2	43.3	53.2	53.3
ClinicalBERT	81.0	85.6	**43.9**	54.3	54.5
GatedCNN-NCI	**83.7**	**87.7**	42.9	**54.4**	**56.6**

4.6 Model Size

We compare the number of trainable parameters (Table 5) of our model with two models with quali-

[4] https://github.com/huggingface/transformers

fied performance, i.e., CAML (Mullenbach et al., 2018) and MultiResCNN (Li and Yu, 2020). HyperCore (Cao et al., 2020) didn't publish the code or provide the values of all hyperparameters. Thus, we omit it in this comparison. Our proposed model is more efficient than the MultiResCNN in terms of the number of trainable parameters. The CAML model has the fewest parameters but performs poorly in prediction. Our model has a much better predictive performance than the CAML model, with only a slight increase in model size.

Table 5: Number of trainable parameters

Model	num. params.
CAML (Mullenbach et al., 2018)	6.2M
MultiResCNN (Li and Yu, 2020)	11.9M
ClinicalBERT (Alsentzer et al., 2019)	113.8M
GatedCNN-NCI (Ours)	7.6M

4.7 Ablation Study

We further conduct an ablation study the investigate the effectiveness of different components of our proposed model. We evaluate two variants by removing two critical components of the proposed model. The first variant without NCI replaces the note-code interaction with max-pooling and linear projection. The second variant removes the gating mechanism that controls the information prorogation over the CNN layers. Table 6 compares the experimental results on the MIMIC-III subset with top-50 codes. The performance drops to some extent after removing these two modules, which shows the effectiveness of our proposed architectures. Moreover, the note-code interaction module has slightly more contribution than the gating mechanism. Possible explanations are that the explicit interaction perseveres the semantics of medical codes well and captures the relation between codes and notes in the embedding space.

Table 6: Ablation study

Model	AUC-ROC		F1		
	Macro	Micro	Macro	Micro	P@5
GatedCNN-NCI	91.5	93.8	62.9	68.6	65.3
without NCI	90.1	92.7	61.4	67.2	63.9
without gating	90.0	92.0	60.2	66.9	63.7

4.8 Case Study

We conduct a case study to interpret an example prediction. Table 7 shows the predictions for a clinical note of a patient with cardiovascular diseases and diabetes. The patient also had 'dyspnea on exertion' as a symptom caused by either pneumonia or cardiac diseases. Our model and MultiResCNN predict the correct diagnosis codes: coronary atherosclerosis (ICD code: 414.01), hypertension (401.9), and diabetes (250.00). When predicting procedure codes, MultiResCNN is confused by dyspnea on exertion and incorrectly predicts pneumonia-related treatments: endotracheal intubation (96.04) and invasive mechanical ventilation (96.71). Our model correctly predicts a cardiac catheterization procedure and diagnostic interventions of heart surgery (39.61) and coronary artery bypass (36.15). Hence, our model is not misled by the ambiguous interpretation for dyspnea on exertion but learns the correct cardiac-related context, consistent with the rest of the note.

Table 7: Case study on a clinical note with cardiac-related diseases (**bold**, in green). Dyspnea on exertion (*italic*, in red) can be caused by cardiac- or pneumonia-related diseases.

Clinical note	old male with multiple **cardiac risk factors** and *dyspnea on exertion* . . . , he then underwent further workup which included **a cardiac catheterization** that revealed significant **coronary artery disease**. he was then transferred for surgical evaluation".	
Prediction	Procedure codes	Diagnosis codes
Gold ICD codes	**36.15**; **39.61**;	401.9; 414.01; 250.00
MultiResCNN	*96.04*; *96.71*;	401.9; 414.01; 250.00
GatedCNN-NCI	**36.15**; **39.61**;	401.9; 414.01; 250.00

5 Conclusion

Medical code assignment from clinical notes is a fundamental task for healthcare information systems and diagnosis decision support. This paper proposes a novel framework with gated convolutional neural networks and note-code message passing mechanism for automated medical code assignment. Our solution can learn meaningful features from lengthy clinical documents and effectively control the deep propagation of information flow. Moreover, the message passing mechanism can enhance the ICD code space's semantics and model the note-code interaction to improve medical code prediction. Experiments show the effectiveness of our proposed method.

Acknowledgments

This work was supported by the Academy of Finland (grant 336033) and EU H2020 (grant

101016775). We acknowledge the computational resources provided by the Aalto Science-IT project. The authors wish to acknowledge CSC - IT Center for Science, Finland, for computational resources.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv preprint arXiv:1803.01271*.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2019. Trellis networks for sequence modeling. In *International Conference on Learning Representations*.

Tian Bai and Slobodan Vucetic. 2019. Improving Medical Code Prediction from Clinical Text via Incorporating Online Knowledge Sources. In *The World Wide Web Conference*, pages 72–82.

Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. 2018. Multi-label Classification of Patient Notes: Case Study on ICD Code Assignment. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and Co-graph Representation for Automatic ICD Coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019. Explicit interaction model towards text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6359–6366.

Richárd Farkas and György Szarvas. 2008. Automatic Construction of Rule-based ICD-9-CM Coding Systems. In *BMC Bioinformatics*, volume 9(Suppl 3), pages 1–9. Springer.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2020. Dilated Convolutional Attention Network for Medical Code Assignment from Clinical Text. In *3rd Clinical Natural Language Processing Workshop at EMNLP*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data*, 3:160035.

Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An Empirical Evaluation of Supervised Learning Approaches in Assigning Diagnosis Codes to Electronic Medical Records. *Artificial Intelligence in Medicine*, 65(2):155–166.

Yoon Kim. 2014. Convolutional Ceural Networks for Sentence Classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *EMNLP*.

Fei Li and Hong Yu. 2020. ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Min Li, Zhihui Fei, Min Zeng, Fang-Xiang Wu, Yaohang Li, Yi Pan, and Jianxin Wang. 2018. Automated ICD-9 Coding via a Deep Learning Approach. *IEEE/ACM transactions on Computational Biology and Bioinformatics*, 16(4):1193–1202.

Lucian Vlad Lita, Shipeng Yu, Stefan Niculescu, and Jinbo Bi. 2008. Large Scale Diagnostic Code Classification for Medical Patient Records. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Julia Medori and Cédrick Fairon. 2010. Machine Learning and Features Selection for Semi-automatic ICD-9-CM Encoding. In *Proceedings of the NAACL*

HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents, pages 84–89. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of NAACL-HLT*, pages 1101–1111.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499*.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis Code Assignment: Models and Evaluation Metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237.

Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2017. Condensed Memory Networks for Clinical Diagnostic Inferencing. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. 2011. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A Public-access Intensive Care Unit Database. *Critical Care Medicine*, 39(5):952.

Elyne Scheurwegs, Kim Luyckx, Léon Luyten, Walter Daelemans, and Tim Van den Bulcke. 2016. Data Integration of Structured and Unstructured Sources for Assigning Clinical Codes to Patient Stays. *Journal of the American Medical Informatics Association*, 23(e1):e11–e19.

Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards Automated ICD Coding Using Deep Learning. *arXiv preprint arXiv:1711.04075*.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. 2017. Breaking the softmax bottleneck: A high-rank RNN language model. In *ICLR*.