A Label Attention Model for ICD Coding from Clinical Text

Thanh Vu[1] , **Dat Quoc Nguyen**[2] and **Anthony Nguyen**[1]
[1]Australian e-Health Research Centre, CSIRO, Brisbane, Australia
[2]VinAI Research, Hanoi, Vietnam
thanh.vu@csiro.au, v.datnq9@vinai.io, anthony.nguyen@csiro.au

Abstract

ICD coding is a process of assigning the **I**nternational **C**lassification of **D**isease diagnosis codes to clinical/medical notes documented by health professionals (e.g. clinicians). This process requires significant human resources, and thus is costly and prone to error. To handle the problem, machine learning has been utilized for *automatic* ICD coding. Previous state-of-the-art models were based on convolutional neural networks, using a single/several fixed window sizes. However, the lengths and interdependence between text fragments related to ICD codes in clinical text vary significantly, leading to the difficulty of deciding what the best window sizes are. In this paper, we propose a new label attention model for automatic ICD coding, which can handle both the various lengths and the interdependence of the ICD code related text fragments. Furthermore, as the majority of ICD codes are not frequently used, leading to the extremely imbalanced data issue, we additionally propose a hierarchical joint learning mechanism extending our label attention model to handle the issue, using the hierarchical relationships among the codes. Our label attention model achieves new state-of-the-art results on three benchmark MIMIC datasets, and the joint learning mechanism helps improve the performances for infrequent codes.

1 Introduction

International Classification of Diseases (ICD) is the global health care classification system consisting of metadata codes.[1] ICD coding is the process of assigning codes representing diagnoses and procedures performed during a patient visit using the patient's visit data, such as the clinical/medical notes documented by health professionals. ICD codes can be used for both clinical research and healthcare purposes, such as for epidemiological studies and billing of services [O'malley *et al.*, 2005; Nguyen *et al.*, 2018].

Manual ICD coding performed by clinical coders relies on manual inspections and experience-based judgment. The effort required for coding is thus labor and time intensive and prone to human errors [O'malley *et al.*, 2005; Nguyen *et al.*, 2018]. As a result, machine learning has been utilized to help automate the ICD coding process. This includes both conventional machine learning [Perotte *et al.*, 2013; Koopman *et al.*, 2015] and deep learning [Karimi *et al.*, 2017; Prakash *et al.*, 2017; Baumel *et al.*, 2018; Mullenbach *et al.*, 2018; Wang *et al.*, 2018; Song *et al.*, 2019; Xie *et al.*, 2019; Li and Yu, 2020]. Automatic ICD coding is challenging due to the large number of available codes, e.g. ∼17,000 in ICD-9-CM and ∼140,000 in ICD-10-CM/PCS,[2] and the problem of highly long tailed codes, in which some codes are frequently used but the majority may only have a few instances due to the rareness of diseases [Song *et al.*, 2019; Xie *et al.*, 2019].

Previous state-of-the-art (SOTA) models on the benchmark MIMIC datasets [Lee *et al.*, 2011; Johnson *et al.*, 2016] were based on convolutional neural networks (CNNs) with single or several fixed window sizes [Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020]. However, the lengths and interdependence of text fragments in clinical documentation related to ICD codes can vary significantly. For example, to identify the ICD code "V10.46: Personal history of malignant neoplasm of prostate" from the clinical text "*...past medical history* asthma/copd, htn, *...prostate cancer...*", we need to highlight both the "past medical history" and "prostate cancer" fragments which are far from each other in the text. Although densely connected CNN [Xie *et al.*, 2019] and multi-filter based CNN [Li and Yu, 2020] could handle the different sizes of a *single* text fragment, selecting optimal window sizes of the CNN-based models for interdependent fragments with different lengths is challenging.

Our contributions. As the *first contribution*, we propose a label attention model for ICD coding which can handle the various lengths as well as the interdependence between text fragments related to ICD codes. In our model, a bidirectional Long-Short Term Memory (BiLSTM) encoder is utilized to capture contextual information across input words in a clinical note. A new label attention mechanism is proposed by extending the structured self-attention mechanism [Lin *et al.*, 2017] to learn label-specific vectors that represent the important clinical text fragments relating to certain labels. Each label-specific vector is used to build a binary classifier for a given label. As the *second contribution*, we additionally propose a hierarchical joint learning mechanism that extends our label attention model to handle the highly imbalanced

[1] https://www.who.int/classifications/icd/factsheet/en/

[2] https://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm

data problem, using the hierarchical structure of the ICD codes. As our *final contribution*, we extensively evaluate our models on three standard benchmark MIMIC datasets [Lee *et al.*, 2011; Johnson *et al.*, 2016], which are widely used in automatic ICD coding research [Perotte *et al.*, 2013; Prakash *et al.*, 2017; Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020]. Experimental results show that our model obtains the new SOTA performance results across evaluation metrics. In addition, our joint learning mechanism helps improve the performances for infrequent codes.

2 Related Work

Automatic ICD coding has been an active research topic in the healthcare domain for more than two decades [Larkey and Croft, 1996; de Lima *et al.*, 1998]. Many conventional machine learning and deep learning approaches have been explored to automatically assign ICD codes on clinical text data, in which the coding problem is formulated as a multi-label classification problem [Perotte *et al.*, 2013; Koopman *et al.*, 2015; Karimi *et al.*, 2017; Shi *et al.*, 2017; Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020].

Larkey and Croft [1996] proposed an ensemble approach combining three feature-based classifiers (i.e., K nearest neighbors, relevance feedback, and Bayesian independence) to assign ICD-9 codes to inpatient discharge summaries. They found that combining the classifiers performed much better than individual ones. de Lima *et al.* [1998] utilized the cosine similarity between the medical discharge summary and the ICD code description to build the classifier which assigns codes with the highest similarities to the summary. They also proposed a hierarchical model by utilizing the hierarchical relationships among the codes. Similarly, Perotte *et al.* [2013] explored support vector machine (SVM) to build flat and hierarchical ICD code classifiers and applied to discharge summaries from the MIMIC-II dataset [Lee *et al.*, 2011]. Apart from discharge summaries, Koopman *et al.* [2015] proposed a hierarchical model of employing SVM to assign cancer-related ICD codes to death certificates. Karimi *et al.* [2017] utilized classification methods for ICD coding from radiology reports.

Deep learning models have been proposed to handle the task recently. Shi *et al.* [2017] employed character-level LSTM to learn the representations of specific subsections from discharge summaries and the code description. They then applied an attention mechanism to address the mismatch between the subsections and corresponding codes. Wang *et al.* [2018] proposed a joint embedding model, in which the labels and words are embedded into the same vector space and the cosine similarity between them is used to predict the labels. Mullenbach *et al.* [2018] proposed a convolutional attention model for ICD coding from clinical text (e.g. discharge summaries). The model is the combination of a single filter CNN and label-dependent attention. Xie *et al.* [2019] improved the convolutional attention model [Mullenbach *et al.*, 2018] by using densely connected CNN and multi-scale feature attention. Graph convolutional neural network [Kipf and Welling, 2017] was employed as the model regularization to capture the hierarchical relationships among the codes.

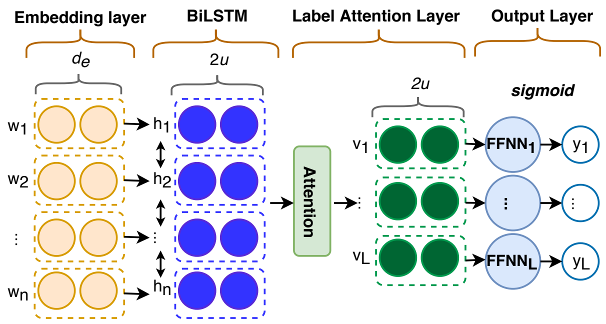

Figure 1: Architecture of our label attention model which contains an embbedding layer, a Bidirectional LSTM layer, a label attention layer and an output layer.

Li and Yu [2020] later proposed a multi-filter residual CNN combining a multi-filter convolutional layer and a residual convolutional layer to improve the convolutional attention model [Mullenbach *et al.*, 2018]. See Section 4.4 of baseline models for additional information.

3 Approach

In this section, we first describe our new **la**bel **at**tention model (namely, **LAAT**) for ICD coding from clinical text. As most of ICD codes do not frequently occur in clinical text data [Koopman *et al.*, 2015; Xie *et al.*, 2019],[3] we additionally propose a hierarchical joint learning mechanism to improve the performance of predicting less-frequent ICD codes.

We treat this ICD coding task as a multi-label classification problem [McCallum, 1999]. Following Mullenbach *et al.* [2018], our objective is to train $|\mathbf{L}|$ binary classifiers (here, $\mathbf{L}$ is the ICD code set), in which each classifier is to determine the value of $y_j \in \{0, 1\}$, the j^{th} label in $\mathbf{L}$ given an input text.

3.1 Our Label Attention Model

Figure 1 illustrates the architecture of our proposed label attention model. Overall, the model consists of four layers. The first layer is an embedding layer in which pretrained word embeddings are employed to produce embedding vectors of tokens in the input clinical text. The second layer is a bidirectional Long Short-Term Memory (LSTM) network producing latent feature representations of all the input tokens. Given these latent representations, the third layer is an attention one producing label-specific weight vectors each representing the whole input text. The last layer consists of label-specific binary classifiers on top of the corresponding label-specific vectors. Each classifier uses a single feed-forward network (FFNN) to predict whether a certain ICD code is assigned to the input text or not.

Embedding Layer

Assume that a clinical document D consists of n word tokens $w_1, w_2, ..., w_i, ..., w_n$. We represent each i^{th} token w_i in D by a pre-trained word embedding $\boldsymbol{e}_{w_i}$ having the same embedding size of d_e.

[3] 5,411 (60%) of all the 8,929 ICD codes appear less than 10 times in the MIMIC-III dataset [Johnson *et al.*, 2016].

Bidirectional LSTM Layer

We use a BiLSTM architecture to capture contextual information across input words in D. In particular, we use the BiLSTM to learn latent feature vectors representing input words from a sequence $\boldsymbol{e}_{w_1:w_n}$ of vectors $\boldsymbol{e}_{w_1}, \boldsymbol{e}_{w_2}, ..., \boldsymbol{e}_{w_n}$. We compute the hidden states of the LSTMs corresponding to the i^{th} word ($i \in \{1, \ldots, n\}$) as:

$$\overrightarrow{\boldsymbol{h}_i} = \overrightarrow{\text{LSTM}}(\boldsymbol{e}_{w_1:w_i}) \quad (1)$$

$$\overleftarrow{\boldsymbol{h}_i} = \overleftarrow{\text{LSTM}}(\boldsymbol{e}_{w_i:w_n}) \quad (2)$$

where $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$ denote forward and backward LSTMs, respectively. Two vectors $\overrightarrow{\boldsymbol{h}_i}$ and $\overleftarrow{\boldsymbol{h}_i}$ are then concatenated to formulate the final latent vector $\boldsymbol{h}_i$:

$$\boldsymbol{h}_i = \overrightarrow{\boldsymbol{h}_i} \oplus \overleftarrow{\boldsymbol{h}_i} \quad (3)$$

The dimensionality of the LSTM hidden states is set to u, resulting in the size of the latent vectors $\boldsymbol{h}_i$ at $2u$. All the hidden state vectors of words in D are concatenated to formulate a matrix $\mathbf{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n] \in \mathbb{R}^{2u \times n}$.

Attention Layer

As the clinical documents have different lengths and each document has multi-labels, our goal is to transform $\mathbf{H}$ into label-specific vectors. We achieve that goal by proposing a label attention mechanism. Our label attention mechanism takes $\mathbf{H}$ as the input and output $|\mathbf{L}|$ label-specific vectors representing the input document D. First, we compute the label-specific weight vectors as:

$$\mathbf{Z} = \tanh(\mathbf{WH}) \quad (4)$$

$$\mathbf{A} = \text{softmax}(\mathbf{UZ}) \quad (5)$$

Here, $\mathbf{W}$ is a matrix $\in \mathbb{R}^{d_a \times 2u}$, in which d_a is a hyperparameter to be tuned with the model, resulting in a matrix $\mathbf{Z} \in \mathbb{R}^{d_a \times n}$. The matrix $\mathbf{Z}$ is used to multiply with a matrix $\mathbf{U} \in \mathbb{R}^{|\mathbf{L}| \times d_a}$ to compute the label-specific weight matrix $\mathbf{A} \in \mathbb{R}^{|\mathbf{L}| \times n}$, in which each i^{th} row of $\mathbf{A}$ refers to as a weight vector regarding the i^{th} label in $\mathbf{L}$. softmax is applied at the row level to ensure that the summation of weights in each row is equal to 1. After that, the attention weight matrix $\mathbf{A}$ is then multiplied with the hidden state matrix $\mathbf{H}$ to produce the label-specific vectors representing the input document D as:

$$\mathbf{V} = \mathbf{HA}^\top \quad (6)$$

Each i^{th} column $\mathbf{v}_i$ of the matrix $\mathbf{V} \in \mathbb{R}^{2u \times |\mathbf{L}|}$ is a representation of D regarding the i^{th} label in $\mathbf{L}$.

Output Layer

For each label-specific representation $\mathbf{v}_i$, we pass it as input to a corresponding single-layer feed-forward network (FFNN) with a one-node output layer followed by a sigmoid activation function to produce the probability of the i^{th} label given the document. Here, the probability is then used to predict the binary output $\in \{0, 1\}$ using a predefined threshold, such as 0.5. The training objective is to minimize the binary cross-entropy loss between the predicted label $\overline{y}$ and the target y as:

$$\text{Loss}(D, y, \theta) = \sum_{j=1}^{|\mathbf{L}|} y_j \log \overline{y}_j + (1 - y_j) \log(1 - \overline{y}_j) \quad (7)$$

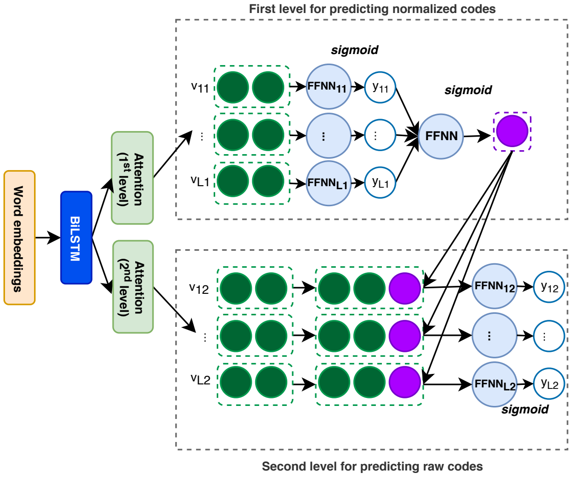

Figure 2: The architecture of our hierarchical joint learning model JointLAAT has two levels: The first level is to predict the normalized codes composing of the first three characters of raw ICD codes. The second level utilizes the prediction produced from the first level to predict the raw ICD codes.

Where θ denotes all the trainable parameters.

Discussion

Our attention layer can be viewed as an extension of the structured self-attention mechanism proposed by Lin *et al.* [2017] for the multi-label classification task. In particular, different from Lin *et al.* [2017], the number of attention hops is set to the number of labels; and we then use the document embedding from each hop separately to build a binary classifier for a certain label. Note that Lin *et al.* [2017] create a single final text embedding aggregated from all the attention hops to make the classification prediction. The approach of using a single aggregated text embedding is suitable for single-label classification problems, such as sentiment analysis [Lin *et al.*, 2017], but not suitable for multi-label text classification tasks, such as ICD coding.

3.2 Hierarchical Joint Learning Mechanism

A challenge of the ICD coding task is that most of the ICD codes are not frequently used leading to an extremely unbalanced set of codes [Song *et al.*, 2019; Xie *et al.*, 2019]. As there are hierarchical relationships between ICD codes, in which codes starting with the same first three characters belong to the same higher-order category, we can utilize the hierarchical structure among the codes to help the model work better for infrequent codes. For example, "Nonpyogenic meningitis" (322.0), "Eosinophilic meningitis" (322.1), "Chronic meningitis" (322.2), "Meningitis, unspecified" (322.9) belong to a category of "Meningitis of unspecified cause" (322).

To this end, we propose a hierarchical joint learning model (namely **JointLAAT**) based on our label attention model, as detailed in Figure 2. For each input document D, the model firstly produces the prediction for the first level of the ICD codes' first three characters (i.e. normalized codes). The predicted output of the first level "normalization" is embedded into a vector $\boldsymbol{s}_D \in \mathbb{R}^p$ with the projection size p. The vector

s_D is then concatenated with each label-specific vector $\mathbf{v}_{i2}$ of the second level of the "raw" ICD codes before being fed into the feed-forward network to produce the final prediction. The model is trained by minimizing the sum of the binary cross-entropy losses of the "normalization" and "raw" levels.

4 Experimental Setup

This section details the methodology to evaluate the effectiveness of our model.

4.1 Datasets

We follow recent SOTA work on ICD coding from clinical text [Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020]: using benchmark **M**edical **I**nformation **M**art for **I**ntensive **C**are (MIMIC) datasets MIMIC-III [Johnson *et al.*, 2016] and MIMIC-II [Lee *et al.*, 2011].

MIMIC-III. Following previous work [Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020], we focus on the discharge summaries, which condense all the information during a patient stay into a single document. Each admission was tagged manually by coders with a set of ICD-9 codes describing diagnoses and procedures during the patient stay. In this dataset, there were 52,722 discharge summaries and 8,929 unique codes in total. We conduct the experiments following the previous work [Mullenbach *et al.*, 2018]. For the first experiment of using the full set of codes, the data was split using patient ID so that no patient is appearing in both training and validation/test sets. In particular, there are 47,719 discharge summaries for training, 1,631 for validation and 3,372 for testing. For the second experiment of using the 50 most frequent codes, the resulting subset of 11,317 discharge summaries was obtained, in which there are 8,067 discharge summaries for training, 1,574 for validation and 1,730 for testing. We denote the datasets used in the two settings as **MIMIC-III-full** and **MIMIC-III-50**, respectively.

MIMIC-II. We also conduct experiments on the MIMIC-II dataset, namely **MIMIC-II-full**. Following the previous work [Perotte *et al.*, 2013; Mullenbach *et al.*, 2018; Li and Yu, 2020], 20,533 and 2,282 clinical notes were used for training and testing, respectively (with a total of 5,031 unique codes). From the set of 20,533 clinical notes, we further use 1,141 notes for validation, resulting in only 19,392 notes for training our model.

Preprocessing. Following the previous work [Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020], we tokenize the text and lowercase all the tokens. We remove tokens containing no alphabetic characters such as numbers, punctuations. For a fair comparison, similar to the previous work, on the preprocessed text from the discharge summaries in the MIMIC-III-full dataset, we pre-train word embeddings with the size $d_e = 100$ using CBOW Word2Vec method [Mikolov *et al.*, 2013]. We then utilize the pretrained word embeddings for all experiments on the three MIMIC datasets. As shown in Li and Yu [2020], there were no significant performance differences when truncating the text to a maximum length ranging from 2,500 to 6,500. We, therefore, truncate all the text to the maximum length of 4,000 as in Xie *et al.* [2019] for the fairness and reducing the computational cost.

4.2 Evaluation Metrics

To make a complete comparison with the previous work on ICD coding, we report the results of our proposed model on a variety of metrics, including macro- and micro-averaged F1 and AUC (area under the ROC curve), precision at k (P@k $\in \{5, 8, 15\}$). As detailed in Manning *et al.* [2008], "micro-averaged" pools per-pair of (text, code) decisions, and then computes an effectiveness measure on the pooled data, while "macro-averaged" computes a simple average over all labels. P@k is the precision of the top-k predicted labels with the highest predictive probabilities.

4.3 Implementation and Hyper-parameter Tuning

Implementation. We implement our LAAT and JointLAAT using PyTorch [Paszke *et al.*, 2019]. We train the models with AdamW [Loshchilov and Hutter, 2019], and set its learning rate to the default value of 0.001.[4] The batch size and number of epochs are set to 8 and 50, respectively. We use a learning rate scheduler to automatically reduce the learning rate by 10% if there is no improvement in every 5 epochs. We also implement an early stopping mechanism, in which the training is stopped if there is no improvement of the micro-averaged F1 score on the validation set in 6 continuous epochs. For both LAAT and JointLAAT, we apply a dropout mechanism with the dropout probability of 0.3. Before each epoch, we shuffle the training data to avoid the influence of the data order in learning the models. We choose the models with the highest micro-averaged F1 score over the validation sets to apply to the test sets. Note that we ran our models 10 times with the same hyper-parameters using different random seeds and report the scores averaged over the 10 runs.

Hyper-parameter tuning. For LAAT, we perform a grid search over the LSTM hidden size $u \in \{128, 256, 384, 512\}$ and the projection size $d_a \in \{128, 256, 384, 512\}$, resulting in the optimal values u at 512 and d_a at 512 on the MIMIC-III-full dataset, and the optimal values u at 256 and d_a at 256 on both the MIMIC-III-50 and MIMIC-II-full datasets. For JointLAAT, we employ the optimal hyper-parameters (d_a and u) from LAAT and fix the projection size p at 128.

4.4 Baselines

Our LAAT and JointLAAT are compared against the following recent SOTA baselines, including both conventional machine learning and deep learning models:

LR. **L**ogistic **R**egression was explored for ICD coding on the MIMIC datasets by building binary one-versus-rest classifiers with unigram bag-of-word features for all labels appearing in the training data [Mullenbach *et al.*, 2018].

SVM. Perotte *et al.* [2013] utilized the hierarchical nature of ICD codes to build hierarchical classifiers using **S**upport **V**ector **M**achine (SVM). Experiments on the MIMIC-II-full dataset showed that hierarchical SVM performed better than the flat SVM which treats the ICD codes independently. Xie *et al.* [2019] applied the hierarchical SVM for ICD coding

[4] In preliminary experiments, we find that though AdamW and Adam [Kingma and Ba, 2015] produce similar performances, AdamW converges faster than Adam when training our models.

on the MIMIC-III-full dataset using 10,000 unigram features with the tf-idf weighting scheme.

CNN. The one-dimensional **C**onvolutional **N**eural **N**etwork [Kim, 2014] was employed by Mullenbach *et al.* [2018] for ICD coding on the MIMIC datasets.

BiGRU. The **bi**directional **G**ated **R**ecurrent **U**nit [Cho *et al.*, 2014] was utilized by Mullenbach *et al.* [2018] for ICD coding on the MIMIC datasets.

C-MemNN. The **C**ondensed **Mem**ory **N**eural **N**etwork was proposed by Prakash *et al.* [2017], which combines the memory network [Sukhbaatar *et al.*, 2015] with iterative condensed memory representations. This model produced competitive ICD coding results on the MIMIC-III-50 dataset.

C-LSTM-Att. The **C**haracter-aware **LSTM**-based **Att**ention model was proposed by Shi *et al.* [2017] for ICD coding. In the model, LSTM-based language models were utilized to generate the representations of clinical notes and ICD codes, and an attention method was proposed to address the mismatch between notes and codes. The model was employed to predict the ICD codes for the medical notes in the MIMIC-III-50 dataset.

HA-GRU. The **H**ierarchical **A**ttention **G**ated **R**ecurrent **U**nit (HA-GRU) [Yang *et al.*, 2016] was utilized by Baumel *et al.* [2018] for ICD coding on the MIMIC-II dataset.

LEAM. The **L**abel **E**mbedding **A**ttentive **M**odel was proposed by Wang *et al.* [2018] for text classification, where the labels and words were embedded in the same latent space, and the text representation was built using the text-label compatibility, resulting in competitive results on MIMIC-III-50.

CAML. The **C**onvolutional **A**ttention network for **M**ulti-**L**abel classification (CAML) was proposed by Mullenbach *et al.* [2018]. The model achieved high performances on the MIMIC datasets. It contains a single layer CNN [Kim, 2014] and an attention layer to generate label-dependent representation for each label (i.e., ICD code).

DR-CAML. **D**escription **R**egularized CAML [Mullenbach *et al.*, 2018] is an extension of the CAML model, incorporating the text description of each code to regularize the model.

MSATT-KG. The **M**ulti-**S**cale Feature **Att**ention and Structured **K**nowledge **G**raph Propagation approach was proposed by Xie *et al.* [2019] achieving the SOTA ICD coding results on the MIMIC-III-full and MIMIC-III-50 datasets. The model contains a densely connected convolutional neural network which can produce variable n-gram features and a multi-scale feature attention to adaptively select multi-scale features. In the model, the graph convolutional neural network [Kipf and Welling, 2017] is also employed to capture the hierarchical relationships among medical codes.

MultiResCNN. The **Multi**-Filter **Res**idual **C**onvolutional **N**eural **N**etwork was proposed by Li and Yu [2020] for ICD coding achieving the SOTA results on the MIMIC-II-full dataset and in-line SOTA results on the MIMIC-III-full dataset. The model contains a multi-filter convolutional layer to capture various text patterns with different lengths and a residual convolutional layer to enlarge the receptive field.

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
LR	56.1	93.7	1.1	27.2	-	54.2	41.1
SVM	-	-	-	44.1	-	-	-
CNN	80.6	96.9	4.2	41.9	-	58.1	44.3
BiGRU	82.2	97.1	3.8	41.7	-	58.5	44.5
CAML	89.5	98.6	8.8	53.9	-	70.9	56.1
DR-CAML	89.7	98.5	8.6	52.9	-	69.0	54.8
MSATT-KG	**91.0**	**99.2**	**9.0**	**55.3**	-	72.8	58.1
MultiResCNN	**91.0**	98.6	8.5	55.2	-	**73.4**	**58.4**
LAAT	91.9	**98.8**	9.9	**57.5**	**81.3**	**73.8**	**59.1**
JointLAAT	**92.1**	**98.8**	**10.7***	**57.5**	80.6	73.5	59.0

Table 1: Results (in %) on the MIMIC-III-full test set. $*$ indicates that the performance difference between our two models LAAT and JointLAAT is significant ($p < 0.01$, using the Approximate Randomization test). All scores in tables 1, 2 and 3 are reported under the same experimental setup. Baseline scores are from the corresponding model papers as detailed in Section 4.4.

5 Experimental Results

5.1 Main Results

MIMIC-III-full

On the MIMIC-III-full dataset, Table 1 shows the results of the evaluation across all quantitative metrics. Specifically, using an attention mechanism, CAML [Mullenbach *et al.*, 2018] produced better performance than both conventional machine learning models (i.e., LR and SVM) and deep learning models (i.e., CNN, BiGRU). Addressing the fixed window size problem of CAML [Mullenbach *et al.*, 2018], MASATT-KG [Xie *et al.*, 2019] and MultiResCNN [Li and Yu, 2020] achieved better results than CAML with improvements in micro-F1 by 1.4% and 1.3%, respectively. Our label attention model LAAT produces higher results in the macro-AUC, macro-F1, micro-F1, P@8 and P@15 metrics, compared to MASATT-KG [Xie *et al.*, 2019] and MultiResCNN [Li and Yu, 2020], while achieving a slightly lower micro-AUC than that of MSATT-KG. In particular, LAAT improves the macro-AUC by 0.9%, macro-F1 by 0.9%, micro-F1 by 2.2%, P@8 by 0.4% and P@15 by 0.7%. LAAT also produces an impressive P@5 of 81.3%, indicating that on average at least 4 out of the top 5 predicted codes are correct.

Regarding JointLAAT where we utilized the hierarchical structures of ICD codes to improve the prediction of infrequent codes, Table 1 also shows that JointLAAT produces better macro-AUC score and significantly higher macro-F1 score than LAAT with the improvement of 0.8% ($p < 0.01$, using the Approximate Randomization test [Chinchor, 1992] which is a nonparametric significance test suitable for NLP tasks [Dror *et al.*, 2018]). Due to the macro-metrics' emphasis on rare-label performance [Manning *et al.*, 2008], this indicates that JointLAAT does better than LAAT for the infrequent codes (the P@k scores of JointLAAT are slightly lower than those of LAAT but the differences are not significant).

MIMIC-III-50

Table 2 shows results on the MIMIC-III-50 dataset. LAAT outperforms all the baseline models across all the metrics. In particular, compared to the previous SOTA model MSATT-

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
LR	82.9	86.4	47.7	53.3	54.6	-	-
C-MemNN	83.3	-	-	-	42.0	-	-
C-LSTM-Att	-	90.0	-	53.2	-	-	-
CNN	87.6	90.7	57.6	62.5	62.0	-	-
BiGRU	82.8	86.8	48.4	54.9	59.1	-	-
LEAM	88.1	91.2	54.0	61.9	61.2	-	-
CAML	87.5	90.9	53.2	61.4	60.9	-	-
DR-CAML	88.4	91.6	57.6	63.3	61.8	-	-
MSATT-KG	**91.4**	**93.6**	**63.8**	**68.4**	**64.4**	-	-
MultiResCNN	89.9	92.8	60.6	67.0	64.1	-	-
LAAT	**92.5**	**94.6**	**66.6**	71.5	**67.5**	**54.7**	**35.7**
JointLAAT	**92.5**	**94.6**	66.1	**71.6**	67.1	54.6	**35.7**

Table 2: Results on the MIMIC-III-50 test set.

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
LR	69.0	93.4	2.5	31.4	-	42.5	-
SVM	-	-	-	29.3	-	-	-
HA-GRU	-	-	-	36.6	-	-	-
CNN	74.2	94.1	3.0	33.2	-	38.8	-
BiGRU	78.0	95.4	2.4	35.9	-	42.0	-
CAML	82.0	96.6	4.8	44.2	-	52.3	-
DR-CAML	82.6	96.6	4.9	45.7	-	51.5	-
MultiResCNN	**85.0**	**96.8**	**5.2**	**46.4**	-	**54.4**	-
LAAT	86.8	**97.3**	5.9	48.6	64.9	55.0	**39.7**
JointLAAT	**87.1**	97.2	**6.8***	**49.1***	**65.2**	**55.1**	39.6

Table 3: Results on the MIMIC-II-full test set.

KG [Xie *et al.*, 2019], LAAT produces notable improvements of 1.1%, 1.0%, 2.8%, 3.1% and 3.1% in macro-AUC, micro-AUC, macro-F1, micro-F1 and P@5, respectively. From Table 2 , we also find that there is no significant difference between LAAT and JointLAAT regarding the obtained scores. The possible reason is that there is no infrequent codes in this dataset, which results in only 8 out of 40 normalized codes (i.e., three character codes) at the first "normalization" level that are linked to more than one raw ICD codes.

MIMIC-II-full

On the MIMIC-II-full dataset, Table 3 shows that LAAT substantially outperforms all the baseline models. Specifically, the micro-F1 is 12.5% higher than HA-GRU [Baumel *et al.*, 2018] which uses another attention mechanism and GRU for the ICD coding task. LAAT differs from HA-GRU in that our attention mechanism is label-specific. Compared to the previous SOTA model MultiResCNN [Li and Yu, 2020], LAAT improves the macro-AUC, micro-AUC, macro-F1, micro-F1 and P@8 by 1.8%, 0.5%, 0.7%, 2.2% and 0.6%, respectively. Similar to the results on the MIMIC-III-full dataset (Table 1), Table 3 shows that JointLAAT does better on infrequent codes than LAAT on the MIMIC-II-full dataset with the improvement of 0.9% on the macro-F1 ($p < 0.01$).

5.2 Ablation Study

As discussed in Section 3.1, our label attention mechanism extends the self-attention mechanism proposed by Lin *et al.* [2017] for a multi-label classification task. MASATT-KG [Xie *et al.*, 2019] and MultiResCNN [Li and Yu, 2020] used another per-label attention mechanism proposed in CAML by Mullenbach *et al.* [2018], in which the weight vector regarding each label was produced directly using the output of a CNN-based network.

Model	AUC		F1		P@k		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
LAAT	92.6	98.8	8.7	58.1	81.8	74.3	58.4
LAAT$_{\text{CAML}}$	89.5	98.3	4.3	43.8	74.8	65.5	49.9
CAML$_{\text{LAAT}}$	90.5	98.2	7.0	52.9	76.5	69.0	54.3
LAAT$_{\text{GRU}}$	91.5	98.6	7.4	55.2	78.9	71.1	56.0

Table 4: Ablation results on the MIMIC-III-full validation set. LAAT$_{\text{CAML}}$: A LAAT variant using the label attention mechanism proposed in CAML instead of our proposed label attention mechanism. CAML$_{\text{LAAT}}$: We modify CAML to use our label attention mechanism instead of the original one in CAML. LAAT$_{\text{GRU}}$: A LAAT variant using BiGRU instead of BiLSTM to learn latent feature vectors representing input words. All the score differences between LAAT and others are significant ($p < 0.01$).

To better understand the model influences, we performed an ablation study on the *validation* set of the MIMIC-III-full dataset. In particular, for the first setting, namely LAAT$_{\text{CAML}}$, we couple the label attention mechanism proposed by Mullenbach *et al.* [2018] with our BiLSTM encoder. Results of LAAT and LAAT$_{\text{CAML}}$ in Table 4 show that our label attention mechanism does better than the label attention mechanism proposed in CAML by Mullenbach *et al.* [2018].

For the second setting, namely CAML$_{\text{LAAT}}$, we employ our attention mechanism on the output of the CNN network used in CAML. Results of LAAT and CAML$_{\text{LAAT}}$ show that employing BiLSTM helps produce better scores than employing CNN under the same attention mechanism.

We further investigate a variant of LAAT, namely LAAT$_{\text{GRU}}$, using a BiGRU encoder instead of a BiLSTM encoder. Table 4 shows that using BiLSTM helps obtain higher performance than using BiGRU. The reason might be that LSTM with the separate memory cells can theoretically remember longer-term dependencies than GRU, thus LSTM is more suitable for ICD coding from long clinical text, e.g. the discharge summaries which are typically *long*.[5]

6 Conclusions

In this paper, we have presented a label attention model for ICD coding from clinical text. We also extend our model with a hierarchical joint learning architecture to handle the infrequent ICD codes. Experimental results on three standard benchmark MIMIC datasets show that our label attention model obtains new state-of-the-art performance with substantial improvements across various evaluation metrics over competitive baselines. The hierarchical joint learning architecture also helps significantly improve the performances for infrequent codes, resulting in higher macro-averaged metrics.

[5] The number of word tokens per document in the MIMIC datasets is about 1,500 on average and can be greater than 6,500 [Mullenbach *et al.*, 2018; Xie *et al.*, 2019; Li and Yu, 2020].

References

[Baumel *et al.*, 2018] Tal Baumel, Jumana Nassour-Kassis, et al. Multi-label classification of patient notes: case study on ICD code assignment. In *Proceedings of the AAAI Workshop on Health Intelligence*, pages 409–416, 2018.

[Chinchor, 1992] Nancy Chinchor. The statistical significance of the MUC-4 results. In *Proceedings of the Conference on Message Understanding*, pages 30–50, 1992.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, 2014.

[de Lima *et al.*, 1998] Luciano R. S. de Lima, Alberto H. F. Laender, and Berthier A. Ribeiro-Neto. A hierarchical approach to the automatic categorization of medical documents. In *Proceedings of CIKM*, pages 132–139, 1998.

[Dror *et al.*, 2018] Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of ACL*, pages 1383–1392, 2018.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, et al. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[Karimi *et al.*, 2017] Sarvnaz Karimi, Xiang Dai, Hamed Hassanzadeh, and Anthony Nguyen. Automatic diagnosis coding of radiology reports: a comparison of deep learning and conventional classification methods. In *Proceedings of BioNLP*, pages 328–332, 2017.

[Kim, 2014] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of EMNLP*, pages 1746–1751, 2014.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of ICLR*, 2017.

[Koopman *et al.*, 2015] Bevan Koopman, Guido Zuccon, Anthony Nguyen, et al. Automatic ICD-10 classification of cancers from free-text death certificates. *International journal of medical informatics*, 84(11):956–965, 2015.

[Larkey and Croft, 1996] Leah S Larkey and W Bruce Croft. Combining classifiers in text categorization. In *Proceedings of SIGIR*, volume 96, pages 289–297, 1996.

[Lee *et al.*, 2011] Joon Lee, Daniel J Scott, et al. Open-access MIMIC-II database for intensive care research. In *Proceedings of EMBC*, pages 8315–8318, 2011.

[Li and Yu, 2020] Fei Li and Hong Yu. ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network. In *Proceedings of AAAI*, 2020.

[Lin *et al.*, 2017] Zhouhan Lin, Minwei Feng, et al. A Structured Self-Attentive Sentence Embedding. In *Proceedings of ICLR*, 2017.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*, 2019.

[Manning *et al.*, 2008] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[McCallum, 1999] Andrew McCallum. Multi-label text classification with a mixture model trained by EM. In *The AAAI workshop on Text Learning*, pages 1–7, 1999.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, et al. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, pages 3111–3119. 2013.

[Mullenbach *et al.*, 2018] James Mullenbach, Sarah Wiegreffe, et al. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of NAACL-HLT*, pages 1101–1111, 2018.

[Nguyen *et al.*, 2018] Anthony N Nguyen, Donna Truran, et al. Computer-Assisted Diagnostic Coding: Effectiveness of an NLP-based approach using SNOMED CT to ICD-10 mappings. In *Proceedings of AMIA*, pages 807–816, 2018.

[O'malley *et al.*, 2005] Kimberly J O'malley, Karon F Cook, et al. Measuring diagnoses: ICD code accuracy. *Health services research*, 40(5p2):1620–1639, 2005.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of NIPS*, pages 8024–8035. 2019.

[Perotte *et al.*, 2013] Adler Perotte, Rimma Pivovarov, et al. Diagnosis code assignment: models and evaluation metrics. *JAMIA*, 21(2):231–237, 2013.

[Prakash *et al.*, 2017] Aaditya Prakash, Siyuan Zhao, et al. Condensed memory networks for clinical diagnostic inferencing. In *Proceedings of AAAI*, page 3274–3280, 2017.

[Shi *et al.*, 2017] Haoran Shi, Pengtao Xie, et al. Towards automated ICD coding using deep learning. *arXiv preprint*, arXiv:1711.04075, 2017.

[Song *et al.*, 2019] Congzheng Song, Shanghang Zhang, et al. Generalized Zero-shot ICD Coding. *arXiv preprint arXiv:1909.13154*, 2019.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, arthur szlam, et al. End-To-End Memory Networks. In *Proceedings of NIPS*, pages 2440–2448. 2015.

[Wang *et al.*, 2018] Guoyin Wang, Chunyuan Li, et al. Joint embedding of words and labels for text classification. In *Proceedings of ACL*, pages 2321–2331, 2018.

[Xie *et al.*, 2019] Xiancheng Xie, Yun Xiong, et al. EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation. In *Proceedings of CIKM*, pages 649–658, 2019.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, et al. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.