Automatic ICD coding exploiting discourse structure and reconciled code embeddings

Shurui Zhang[1], **Bozheng Zhang**[1], **Fuxin Zhang**[1], **Bo Sang**[1] **and Wanchun Yang** [*21]
[1]MsunHealth Co., LTD
[2]Shandong Jiaotong University
{zhang65821, oscarzbz, zfx050621, kangtasz}@gmail.com
yangwanchun82@gmail.com

Abstract

The International Classification of Diseases (ICD) is the foundation of global health statistics and epidemiology. The ICD is designed to translate health conditions into alphanumeric codes. A number of approaches have been proposed for automatic ICD coding, since manual coding is labor-intensive and there is a global shortage of healthcare workers. However, existing studies did not exploit the discourse structure of clinical notes, which provides rich contextual information for code assignment. In this paper, we exploit the discourse structure by leveraging section type classification and section type embeddings. We also focus on the class-imbalanced problem and the heterogeneous writing style between clinical notes and ICD code definitions. The proposed reconciled embedding approach is able to tackle them simultaneously. Experimental results on the MIMIC dataset show that our model outperforms all previous state-of-the-art models by a large margin. The source code is available at https://github.com/discnet2022/discnet

1 Introduction

The International Classification of Diseases (ICD) is a classification system maintained by the World Health Organization. The system is designed to map health conditions to pre-defined ICD codes, allowing the world to share healthcare data in a consistent way between different regions. The foundation of global health statistics and epidemiology is based on the ICD.

The ICD coding task (as shown in table 1) is usually performed by professional coders. Coders review the whole clinical documents and manually assign the most appropriate codes. However, manual coding is labor-intensive, expensive, and error-prone. The approximate cost of ICD coding is estimated to be about $25 billion per year in the US (Lang, 2007).

Clinical Document	**History of Present Illness**: A 62-year-old male with Type II diabetes mellitus, coronary artery disease, hypertension, chronic kidney disease... **Past Medical History**: Hypertension Type II Diabetes Mellitus *s/p cervical laminoplasty*... **Brief Hospital Course**: ... **Discharge Diagnosis**: Anasarca Heart failure with restrictive physiology...
Assigned ICD codes	428.31 *Diastolic heart failure* 584.9 *Acute renal failure, unspecified* 427.32 *Atrial flutter* ...

Table 1: An illustration of ICD coding task

In recent years, deep learning approaches have demonstrated promising results on ICD coding. Some of these studies improved clinical document representation by leveraging Convolutional Neural Networks (CNN) (Mullenbach et al., 2018; Xie et al., 2019). The others improved ICD code representation by exploiting the dependencies between codes (Xie and Xing, 2018; Vu et al., 2020; Cao et al., 2020). However, these approaches entail limitations. Firstly, they ignore the discourse structure of clinical documents. Secondly, most of these approaches did not consider the writing style discrepancies between ICD code descriptions and relevant clinical documents related to the codes. Thirdly, most of these approaches did not consider the class imbalanced problem of the label spaces.

Why is the discourse structure important? Medical professionals prepare clinical documents in different sections. The sections convey discourse-level information and follow rhetorical moves of argumentation (Teufel et al., 1999). Such as "History of Present Illness", "Past Medical History", followed by "Hospital Course", etc. The health conditions that appear in different sections may contribute differently to code assignments. For

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2883–2891
October 12–17, 2022.

example, in table 1, the *s/p cervical laminoplasty* in the past medical history is not related to the current hospitalization and does not contribute to code assignment. In such cases, omitting discourse-level information may mislead the coding task. The identification of the discourse structure can also benefit word sense disambiguation. For example, the acronym *BS* probably signifies *blood sugar* in the laboratory test section, but more likely signifies *breath sounds* in the physical examination section (Li et al., 2010). Therefore the meaning of a health condition must be considered from a discourse-level point of view.

The heterogeneity between ICD code descriptions and relevant clinical documents. Each ICD code is associated with a code description. For example, the code description of 414.01 is *Coronary atherosclerosis of native coronary artery*. A code description provides a formal definition of an ICD code. On the contrary, clinical documents that are written by physicians usually in an informal way, accompanied with telegraphic phrases and abbreviations. For example, *Coronary artery disease* is denoted by *CAD*. The writing style is highly heterogeneous between the code descriptions and relevant clinical documents.

The class imbalanced problem. Most of the recently proposed methods are based on a per-label attention mechanism that was initially proposed by Mullenbach et al. (2018). In this setting, the attention parameters for each label can be considered as the representation for each ICD code, which are learned from relevant segments in clinical documents (hereinafter referred to as "relevant documents") that are highlighted by the attention mechanism. However, the label frequency follows a highly skewed distribution. About 50% of the codes have less than 5 occurrences. In such a case, it is difficult to learn decent representations for instance-scarce codes. Considering the nature of code descriptions and the label distribution, we argue that instance-scarce code representations are supposed to learn more from code descriptions, since code descriptions are the essential definitions of ICD codes. On the contrary, instance-rich code representations are supposed to learn more from relevant documents, since relevant documents provide various expressions of each code.

In this paper, we design a novel neural architecture for automatic ICD coding given unstructured clinical documents:

- To the best of our knowledge, our work is the first to incorporate discourse-level features into automatic ICD coding. Our proposed **Disc**ourse **N**et (DiscNet) exploits discourse-level features by utilizing section type embeddings. In addition, we combine word-level features and sentence-level features for better expressive power.
- We propose a **R**econciled **E**mbedding (RE) approach to learn ICD code representations, mitigating the class imbalanced problem while reconciling the heterogeneity between code descriptions and relevant clinical documents.
- Experimental results on the MIMIC-III dataset (Johnson et al., 2016) show that our method outperforms all previous state-of-the-art methods across evaluation metrics by a large margin.

2 Related Works

Recently released automatic ICD coding approaches are mainly based on deep learning and performed on unstructured clinical documents. Baumel et al. (2018) proposed a possibility to exploit discourse structure, which inspired our work. Mullenbach et al. (2018) proposed a convolutional attention model and outperformed existing state-of-the-art methods (Baumel et al., 2018). Li and Yu (2020) and Xie et al. (2019) improved the convolutional attention model by exploiting multi-scale features. However, it is challenging for a CNN-based model to capture long-term dependencies in a document.

Discourse analysis is a task to model language phenomena that go beyond the individual sentences (Joty et al., 2019). There are few relevant works that focus on discourse analysis in the clinical domain. Li et al. (2010) focused on the discourse analysis of clinical notes and performed argumentative zoning (Teufel et al., 1999) using a hidden markov model. Denny et al. (2009) leveraged NLP techniques to categorize section headers in clinical documents.

To reconcile the heterogeneous writing styles of diagnosis descriptions and ICD code descriptions, Xie and Xing (2018) proposed an adversarial learning approach, which inspired our work.

Some studies worked on addressing the class imbalanced problem. Mullenbach et al. (2018)

proposed a regularization method using embedded code descriptions to improve the performance on infrequent codes. However, the method worsened the average performance on the MIMIC-III dataset. Some methods improved the performance on infrequent codes by modeling the hierarchical structure of ICD codes (Xie et al., 2019; Vu et al., 2020). Zhou et al. (2021) leveraged an interactive shared representation network to alleviate the long-tail problem.

3 Method

We propose a novel neural architecture for automatic ICD coding given unstructured discharge summaries. A discharge summary from an Electronic Health Record (EHR) is an unstructured clinical document that outlines the details of a hospital stay. We partition clinical documents into sections and exploit the discourse structure by leveraging section type embeddings. The ICD code representations are learned using a reconciled embedding approach. Finally, we use a dot production to predict the codes.

3.1 Discourse Net

Discourse Net (DiscNet) exploits discourse-level features, word-level features, and sentence-level features to learn multi-granularity clinical document representations as shown in Figure 1.

3.1.1 Section Type Embeddings

Clinical documents usually contain multiple sections with nonstandardized section headings. We partition a document into sections by identifying the locations of section headings using regular expressions. Terms that clinicians use to label sections are ambiguous and various, e.g. *past medical history* might appear as *pmh*. Due to various writing conventions, we extracted more than 10,000 distinct headings. We chose the top 100 most frequent headings as known section types since they accounted for 93% of the total heading occurrence. We map each section to known section types using a naive bayes classifier based on TF-IDF vectorized section contents. Concretely, each section content is converted to a TF-IDF vector. Then a naive bayes classifier is trained using known section types as labels. Finally, the trained naive bayes model map each section to known section types. We initialize an embedding matrix for know section types: $\boldsymbol{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{100}\}$. Where each $\boldsymbol{s}$ is a d dimensional vector, representing a known section type.

3.1.2 Input Layer

The input word sequence is mapped into an embedding space using pre-trained word embeddings. The word embeddings of size $d = 100$ are pre-trained on the training set of the MIMIC-III dataset using the word2vec CBOW method (Mikolov et al., 2013). The word embedding sequence is denoted as $\boldsymbol{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{|\boldsymbol{D}|}\}$, where $\boldsymbol{w} \in \mathbb{R}^d$ denotes a word vector, $|\boldsymbol{D}|$ denotes the number of words. The input embeddings are the sum of word embeddings and section type embeddings, which can be denoted as $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_{|\boldsymbol{D}|}\}$. Each $\boldsymbol{e}_i$ is the sum of a word embedding $\boldsymbol{w}_i$ and the associated section type embedding $\boldsymbol{s}_j$.

3.1.3 Multi-Granularity Representations

$\boldsymbol{E}$ is a combination of word-level features and discourse-level features. Besides that, we carry sentence-level features for better expressive power. We use a bidirectional GRU (Chung et al., 2014) to model the sequential structure of $\boldsymbol{E}$:

$$\overrightarrow{\boldsymbol{H}}, \overleftarrow{\boldsymbol{H}} = \text{BiGRU}\left(\boldsymbol{E}\right), \quad \boldsymbol{H} = \left(\overrightarrow{\boldsymbol{H}} \parallel \overleftarrow{\boldsymbol{H}}\right) \boldsymbol{W}_1 \tag{1}$$

Where $\parallel$ denotes concatenation. $\boldsymbol{W}_1 \in \mathbb{R}^{2d \times d}$ is a trainable weight matrix to project the dimensionality of the forward and backward hidden states from $2d$ to d.

Let $\boldsymbol{C} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ denote the ICD code representations obtained through the Reconciled Embedding (RE) approach (refer to subsection 3.2). $|\boldsymbol{C}|$ is the number of distinct ICD codes. $\boldsymbol{C}$ is used as attention parameters to interact with document representations. A per-label attention mechanism (Bahdanau et al., 2014) is applied to re-express a document with respect to each code.

$$\begin{gathered} \boldsymbol{Z} = \boldsymbol{H}\boldsymbol{C}^{\mathrm{T}}, \quad \boldsymbol{\alpha} = \text{Softmax}_1\left(\boldsymbol{Z}\right) \\ \boldsymbol{V}^{\text{word}} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H} \end{gathered} \tag{2}$$

T denotes matrix transposition. $\boldsymbol{\alpha} \in \mathbb{R}^{|\boldsymbol{D}| \times |\boldsymbol{C}|}$ are attention weights of a document representation associated with each code. Softmax_1 is applied to the first dimension of $\boldsymbol{Z}$, ensuring the distribution over locations in a document sum to 1. $\boldsymbol{V}^{\text{word}} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ is the code-specific document representations at word level.

We concatenate the hidden state of $\overrightarrow{\boldsymbol{H}}$ at the end position of a sentence and the hidden state of $\overleftarrow{\boldsymbol{H}}$ at the start position of a sentence to embed a sentence.

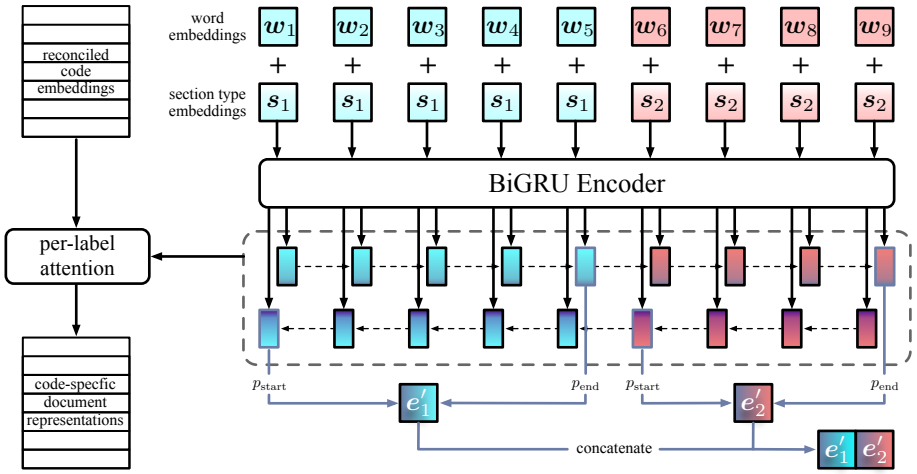

Figure 1: An illustration of how DiscNet works. The sentence embeddings $\boldsymbol{e}'_1$ and $\boldsymbol{e}'_2$ correspond to the input word embeddings $\{\boldsymbol{w}^l_1, ..., \boldsymbol{w}^l_5\}$ and $\{\boldsymbol{w}^l_6, ..., \boldsymbol{w}^l_9\}$ respectively.

Then we concatenate all sentence embeddings:

$$\begin{aligned} \boldsymbol{e}' &= \left(\overrightarrow{\boldsymbol{H}}[p_{\text{end}}] \parallel \overleftarrow{\boldsymbol{H}}[p_{\text{start}}]\right) \boldsymbol{W}_2 \\ \boldsymbol{E}' &= \boldsymbol{e}'_1 \parallel \boldsymbol{e}'_2 \parallel ... \parallel \boldsymbol{e}'_{|P|} \end{aligned} \tag{3}$$

Where $\boldsymbol{E}'$ represents sentence-level features. $\boldsymbol{W}_2 \in \mathbb{R}^{2d \times d}$ is a weight matrix. $|P|$ denotes the number of sentences. We follow the computation step of equation 1 and equation 2 with newly initialized network parameters to obtain $\boldsymbol{V}^{\text{sen}}$, which is the code-specific document representations at sentence level. Finally, we concatenate $\boldsymbol{V}^{\text{word}}$ and $\boldsymbol{V}^{\text{sen}}$. Then a Max Pooling is applied over the level dimension to obtain the condensed code-specific document representations:

$$\boldsymbol{V} = \text{MaxPooling}\left(\boldsymbol{V}^{\text{word}} \parallel \boldsymbol{V}^{\text{sen}}\right) \tag{4}$$

$\boldsymbol{V} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ is the condensed code-specific document representations.

3.2 Reconciled Embedding (RE)

We focus on the class-imbalanced problem and the heterogeneity between code definitions and relevant documents. The proposed RE approach is designed to reconcile them simultaneously.

3.2.1 Reconciling the heterogeneity

We initialize a new bidirectional GRU to encode the word embeddings of code descriptions. Similar to equation 1, the final hidden states of both directions are extracted and projected. Let $\tilde{\boldsymbol{C}} = \{\tilde{\boldsymbol{c}}_1, \tilde{\boldsymbol{c}}_2, ..., \tilde{\boldsymbol{c}}_{|\boldsymbol{C}|}\} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ denote the encoded code descriptions. We design a gate mechanism as follows:

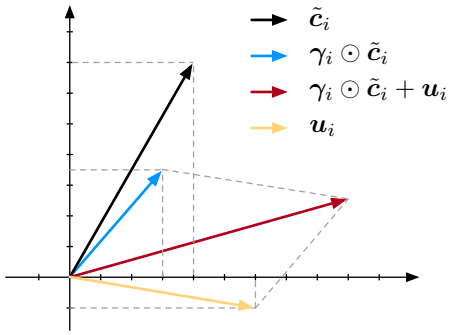

Figure 2: An example of equation 6 in two dimensional space. $\boldsymbol{\gamma}_i$ is set to $[0.75, 0.5]^\top$, $\tilde{\boldsymbol{c}}_i$ is set to $[4.0, 7.0]^\top$, $\boldsymbol{u}_i$ is set to $[6.0, -1.0]^\top$.

$$\begin{aligned} \boldsymbol{Q} &= \text{ReLU}\left(\left(\tilde{\boldsymbol{C}} \parallel \boldsymbol{U}\right) \boldsymbol{W}_3\right) \\ \boldsymbol{\Gamma} &= \text{sigmoid}\left(\boldsymbol{Q}\boldsymbol{W}_4\right) \end{aligned} \tag{5}$$

$\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_{|\boldsymbol{C}|}\} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ is trainable code-specific attention parameters, which are supposed to learn from relevant documents. $\tilde{\boldsymbol{C}}$ and $\boldsymbol{U}$ are concatenated at the last dimension, followed by a linear projection and a ReLU non-linearity. $\boldsymbol{W}_3 \in \mathbb{R}^{2d \times d}$ and $\boldsymbol{W}_4 \in \mathbb{R}^{d \times d}$ are trainable weight matrices. $\boldsymbol{\Gamma} = \{\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, ..., \boldsymbol{\gamma}_{|\boldsymbol{C}|}\} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ is the gate vectors to adjust $\tilde{\boldsymbol{C}}$:

$$\boldsymbol{C} = \text{LayerNorm}\left(\tanh\left(\left(\boldsymbol{\Gamma} \odot \tilde{\boldsymbol{C}} + \boldsymbol{U}\right) \boldsymbol{W}_5\right)\right) \tag{6}$$

Where tanh is the hyperbolic tangent function. LayerNorm denotes layer normalization (Ba et al., 2016). $\odot$ denotes hadamard product. $\boldsymbol{W}_5 \in \mathbb{R}^{d \times d}$ is a trainable weight matrix. $\boldsymbol{C} \in \mathbb{R}^{|\boldsymbol{C}| \times d}$ is the

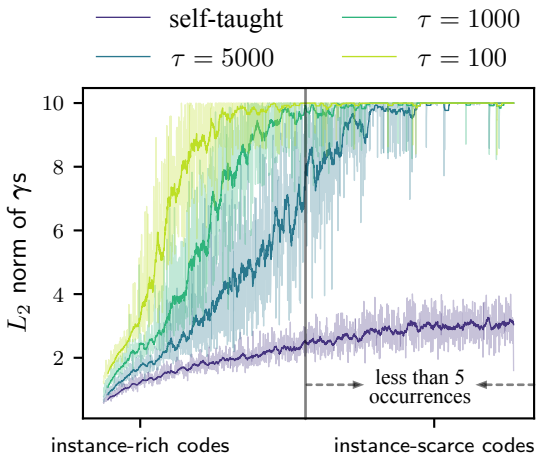

Figure 3: L_2 norm of γs with respect to each code sorted by code occurrences in descending order. A moving average smoothing is applied with a window size equal to 100. A smaller norm indicates less information flow from code descriptions.

reconciled code embeddings to interact with document representations. An example of equation 6 in two dimensional space is given in figure 2.

We suggest the function of the gate mechanism is twofold: Firstly, the element-wise addition of $\tilde{\boldsymbol{C}}$ and $\boldsymbol{U}$ is able to aggregate semantics both from code descriptions and relevant documents. Considering their heterogeneous nature, we leverage the gate vectors $\boldsymbol{\Gamma}$ to adjust $\tilde{\boldsymbol{C}}$, reconciling the semantic discrepancies between $\tilde{\boldsymbol{C}}$ and $\boldsymbol{U}$ during the element-wise addition. Each $\boldsymbol{\gamma}_i$ is able to tune both the length and the direction of $\tilde{\boldsymbol{c}}_i$ within the same quadrant (as shown in figure 2). A naive example is shown in figure 2. Secondly, each $\boldsymbol{\gamma}_i$ is able to scale the information flow from $\tilde{\boldsymbol{c}}_i$. We assume for rare classes, the norm of $\boldsymbol{\gamma}$ is required to be greater than for frequent classes, since it is necessary to learn more from the encoded code definitions for rare classes.

3.2.2 Reconciling the class-imbalance

We assume taking more information from code descriptions than relevant documents could benefit the representation learning of rare codes. But for frequent code, it could be beneficial to take more information from relevant documents than code descriptions. To examine our assumption, we plot the L_2 norm of the trained gate vectors (i.e. $\{\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, ..., \boldsymbol{\gamma}_{|\boldsymbol{C}|}\}$) with respect to code occurrences sorted in descending order as shown in figure 3. The observation supports our assumption. The "self-taught" plot in figure 3 indicates that the proportion of the information transferred is related to the code distribution. The $\boldsymbol{\gamma}$s has learned to transfer more information from code descriptions to rare codes.

To better cope with the class imbalanced problem, we have found imposing a regularization on $\boldsymbol{\gamma}$s according to code distribution to be beneficial. We regularize the L_2 norm of each $\boldsymbol{\gamma}$ according to code distribution. The regularization encourages the norm of rare codes' $\boldsymbol{\gamma}$s to be large, forcing them to learn more from the essential definitions in code descriptions. Meanwhile, there's less regularization imposed on frequent codes since they can learn their heterogeneous expressions from relevant documents. Let $K \in \{k_1, k_2, ..., k_{|\boldsymbol{C}|}\}$ denote the number of occurrences of each code. The regularization term is computed as follows:

$$k'_i = \left(\frac{\max(K) - k_i}{\max(K) - \min(K)} \right)^{\tau}$$
$$L_{reg} = -\sum_{i}^{|C|} k'_i \|\boldsymbol{\gamma}_i\|_2 \tag{7}$$

Firstly, a min-max normalization is applied to rescale the range of K in [0, 1]. k'_i is the regularization weights associated with each code. τ is a rescaling hyperparameter. As shown in figure 3, the smaller τ is, the greater the regularization on frequent codes.

3.3 Output Layer

Firstly, $\boldsymbol{V}$ is fed to a linear layer followed by a ReLU non-linearity. Then a dot product of $\boldsymbol{C}$ and $\boldsymbol{V}$ is applied. Finally, a sigmoid activation function is applied to obtain the probability vector:

$$\boldsymbol{V} = \text{ReLU}(\boldsymbol{V}\boldsymbol{W}_6), \quad \hat{\boldsymbol{y}} = \text{sigmoid}(\boldsymbol{C}\boldsymbol{V}^{\text{T}}) \tag{8}$$

Let $\boldsymbol{y}$ denote the label vector. The code assignment task is treated as a multi-label classification problem. The training objective is to minimize the binary cross-entropy loss and the regularization term from subsection 3.2:

$$\text{Loss}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \text{CrossEntropy}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \lambda L_{reg} \tag{9}$$

Where $\boldsymbol{x}$ denotes input word tokens and $\boldsymbol{\theta}$ denotes all trainable parameters. λ is a hyperparameter.

Model	AUROC		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* (Li and Yu, 2020)	0.910	0.986	0.085	0.552	0.734	0.584
HyperCore* (Cao et al., 2020)	0.930	0.989	0.090	0.551	0.722	0.579
DiscNet+RE* (Ours)	**0.945**	**0.991**	**0.137**	**0.579**	**0.760**	**0.608**
MSATT-KG (Xie et al., 2019)	0.910	0.992	0.090	0.553	0.728	0.581
LAAT (Vu et al., 2020)	0.919	0.988	0.099	0.575	0.738	0.591
JointLAAT (Vu et al., 2020)	0.921	0.988	0.107	0.575	0.735	0.590
ISD (Zhou et al., 2021)	0.938	0.990	0.119	0.559	0.745	-
DiscNet+RE (Ours)	**0.956**	**0.993**	**0.140**	**0.588**	**0.765**	**0.614**

Table 2: Experimental results on the MIMIC-III full test set. Models with "*" are under a length limitation of 2,500. Models without "*" are under a length limitation of 4,000. We ran our model 5 times and averaged the scores.

4 Experiments

4.1 Dataset and Preprocessing

We use publicly available and widely studied MIMIC-III dataset (Johnson et al., 2016), which is an extension of the MIMIC-II dataset (Saeed et al., 2011). The dataset comprises de-identified EHR associated with over 40,000 ICU admissions. We follow the well studied MIMIC-III full setting that was initially proposed by Mullenbach et al. (2018), which consists of 8,929 ICD codes, 47,719, 1,631, and 3,372 discharge summaries for training, development, and testing respectively.

To better investigate the performance of our method on codes with different sample sizes, we divide the test set into head, body, and tail subsets. Each subset has the same number of discharge summaries as in the MIMIC-III full test set but with different range of codes. In the head subset there are 1446 distinct codes with sample size greater than or qual to 50. In the body subset there are 1779 distinct codes with sample size less than 50 and greater than 5. In the tail subset there are 860 distinct codes with sample size less than or equal to 5. There are only 4085 distinct codes in total present in the MIMIC-III full test set.

We tokenize the text, then lowercase and lemmatize the words. All numbers are replaced with a "NUM" token. We perform sentence segmentation using spaCy library[1].

4.2 Evaluation Metrics

For a complete comparison with previous studies, we use macro-averaged and micro-averaged F1, macro-averaged and micro-averaged AUC (area under the receiver operating characteristic curve) and P@k (precision at k).

4.3 Hyper-parameter Tuning and Training

The model is trained using Adam optimizer (Kingma and Ba, 2014) and the initial learning rate is set to 0.0005, the batch size is set to 12. The d-dimensional word embeddings are trainable. A dropout mechanism (Srivastava et al., 2014) is applied after each BiGRU with a dropout probability of 0.2. We notice that the model with section type embeddings is more prone to overfitting. Therefore a dropout mechanism is applied on the section type embeddings with a dropout probability of 0.5. τ is set to 1,000, and λ is set to 0.0001.

4.4 Baselines

Some studies truncated the input text to a maximum length of 2,500, the others to a maximum length of 4,000. We have noticed that there are performance differences between different length limitation settings. For a fair comparison, we conducted the experiments under the length limitations of 2,500 and 4,000, then report the results separately.

CAML: The first per-label attention based model for automatic ICD coding preposed by Mullenbach et al. (2018).

DR-CAML: An extension of CAML which incorporates the code descriptions to improve the performance on rarely observed codes. However, DR-CAML performed worse on most metrics than CAML.

MSATT-KG: The MSATT-KG (Xie et al., 2019) approach employed a graph convolutional neural network to capture the hierarchical relationships among codes, alleviating the class imbalanced problem. The study achieved SOTA performance.

HyperCore: Proposed by Cao et al. (2020), which

[1]https://spacy.io/

Model	AUROC		F1		P@k		Macro F1		
	Macro	Micro	Macro	Micro	8	15	head	body	tail
BiGRU	0.904	0.986	0.097	0.562	0.734	0.581	0.369	0.164	0.052
BiGRU+discourse	0.919	0.988	0.115	0.583	0.752	0.601	0.408	0.211	0.063
only DiscNet	0.919	0.988	0.119	0.583	0.757	0.605	0.419	0.216	0.064
only $\text{RE}_{\text{self-taught}}$	0.942	0.990	0.126	0.567	0.750	0.595	0.400	0.214	0.086
$\text{DiscNet+RE}_{\text{self-taught}}$	0.943	0.990	0.134	0.575	0.756	0.603	0.420	0.235	0.097
$\text{DiscNet+RE}_{\text{constant-}\gamma\text{s}}$	0.938	0.990	0.129	0.574	0.757	0.603	0.419	0.225	0.082
$\text{DiscNet+RE}_{\tau=100}$	**0.947**	**0.991**	0.132	0.578	0.756	0.606	0.420	**0.241**	**0.106**
$\text{DiscNet+RE}_{\tau=1000}$	0.945	**0.991**	**0.137**	**0.579**	**0.760**	**0.608**	**0.425**	0.240	**0.106**
$\text{DiscNet+RE}_{\tau=5000}$	0.946	**0.991**	0.132	**0.579**	0.756	0.604	0.419	0.229	0.100

Table 3: Ablation results on the MIMIC-III full test set under a length limitation of 2500. We ran each model 5 times and averaged the scores.

can jointly exploit code hierarchy and code co-occurrence. The approach outperformed all existing baseline models.

MultiResCNN: The MultiResCNN (Li and Yu, 2020) leveraged a multi-filter convolutional layer to capture various text patterns.

LAAT: A label attention model was proposed by Vu et al. (2020). LAAT outperformed all existing baseline models.

JointLAAT: An extension of LAAT, which leveraged a hierarchical joint learning mechanism to handle the class imbalanced problem.

ISD: The ISD approach (Zhou et al., 2021) leveraged an interactive shared representation network to alleviate the long-tail problem.

4.5 Compared with Baselines

Table 2 show the experimental results on the MIMIC-III full dataset. Our model outperformed all baseline models across all evaluation metrics and achieves new state-of-the-art results. It is worth noting that the macro-AUROC and macro-F1 were improved by 1.8% and 2.1% compared with the best baseline model. The improvements indicate our model is more robust to infrequent code assignments, since macro-averaging highlights the performance of infrequent classes. Meanwhile, the micro-F1, P@8, and P@15 were improved by 2.9%, 2%, and 2.4% respectively. The results suggest our model improves both macro-averaging and micro-averaging measurements simultaneously.

4.6 Ablation study

We performed an ablation study as shown in table 3 and in figure 4. In order to investigate the effectiveness of our methods on codes with different sample sizes, we use macro-F1 to evaluate each ablation experiment on the head, body, tail subsets of the MIMIC-III test set (refer to subsection 4.1)[2]. For the BiGRU setting, we use a BiGRU to model the input word embeddings, followed by a per-label attention to perform classification. For the BiGRU+discourse setting, we add section type embeddings to the BiGRU setting, achieving 1.8% improvement on Macro-F1 and 2.1% on Micro-F1, significantly improved Macro-F1 on the head subset by 3.9%. The improvements demonstrate the effectiveness of exploiting discourse-level features. For the only DiscNet setting, we add sentence-level features to the BiGRU+discourse setting, The minor improvements indicate the effectiveness of sentence-level features. The only

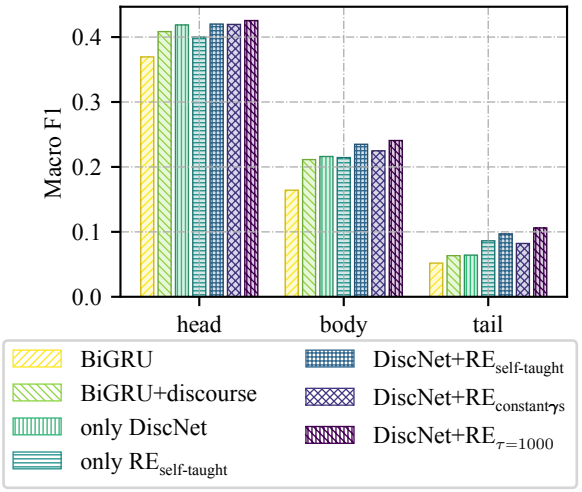

Figure 4: Ablation experiments on head, body and tail subsets demonstrate the improvements of DiscNet on frequent codes and the RE approach on infrequent codes.

[2]In the data splitting setting of Mullenbach et al. (2018), there are only 4,085 distinct codes out of 8,929 present in the test set. The F1 score of a non-appearing code is evaluated to 0. To better compare the F1 score of the head, body, and tail subsets. We evaluate only the 4,085 codes that are present in the test set.

	with section type embeddings	w/o section type embeddings
Case 1	**Brief Hospital Course**: #VB: She had *increased bleeding* ... leading up to presentation... Anemia: Ms. On HD#6, her *hematocrit trended down* to 25 ... ... was transfused 2 units of *red blood cells*... **Discharge Diagnosis**: *Anemia*...	**Past Medical History**: 1.Uterine fibroids 2.*Anemia*, iron-deficiency... **History of Present Illness**: ... with history of *anemia* secondary... **Brief Hospital Course**: ... her *hematocrit trended down* to 25... **Discharge Diagnosis**: *Anemia*...
	285.1 *Acute posthemorrhagic anemia*	~~285.9 *Unspecified anemia*~~
Case 2	**"self-taught"**	**constant γs**
	Smoked 30 yrs, 2 ppd, quit on... ... Patient recently *quit smoking*... We encouraged to continue *smoking abstinence*... He *quit smoking* two week ago due to shortness of breath...	We continued Lisinopril 5mg PO daily... Spiculated pulmonary lesions: consider infectious/inflammatory/neoplastic... In the right, pulmonary artery embolus...
	V15.82 prediction score: 0.70 *personal history of tobacco use*	V15.82 prediction score: 0.02 *personal history of tobacco use*

Table 4: Each example contains a predicted ICD code and relevant document with high attention score.

$\text{RE}_{\text{self-taught}}$ setting, namely the "self-taught" model in figure 3, achieving 3.8% improvement on Macro-AUC, 2.9% on Macro-F1 and 3.4% on Macro-F1 on the tail subset compared to the BiGRU model. The significant improvements demonstrate the effectiveness of RE, particularly on rare codes. For DiscNet+$\text{RE}_{\text{constant}\gamma\text{s}}$ setting, all γs are set equal to 1. The performance drop indicates the effectiveness of the gate mechanism in 3.2.1. We experimented with different τ values and $\tau = 1000$ yields the best results, which demonstrates the effectiveness of the regularization approach in 3.2.2.

4.7 Case Study

To better understand the effectiveness of our approaches, we give examples shown in table 4. We investigate the relevant documents with high attention scores associated with a predicted ICD code. For the first case, the type of *Anemia* is not specified in the discharge diagnosis. The model with section type embeddings correctly linked *bleeding* in the "Brief Hospital Course" section to *anemia*. On the contrary, the unspecified *Anemia* that appears in the "Past Medical History" and the "history of present illness" mislead the baseline model to *285.9 unspecified anemia*. The second example illustrates the impacts of the heterogeneity between code descriptions and relevant documents. The code description *personal history of tobacco use* and relevant document (*smoked, smoking abstinence, and etc*) are literally very different. The "self-taught" model has successfully linked them together. In contrast, the model with constant γs, namely all γs are set equal to 1, failed to highlight any meaningful relevant document.

5 Conclusion

This paper proposed a novel neural architecture for automatic ICD coding. We leverage section type embeddings to make our model discourse-aware. We focus on the class imbalanced problem and the heterogeneity between code definitions and relevant documents. The proposed Reconciled Embedding approach tackled them simultaneously. We achieve state-of-the-art performance on the widely-studied MIMIC-III dataset. DiscNet can be applied to all texts with a discourse structure, but not limited to clinical texts. The proposed reconciled embedding approach can be applied in scenarios where there is auxiliary information associated with labels.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie 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.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Joshua C Denny, Anderson Spickard III, Kevin B Johnson, Neeraja B Peterson, Josh F Peterson, and Randolph A Miller. 2009. Evaluation of a method to identify and categorize section headers in clinical documents. *Journal of the American Medical Informatics Association*, 16(6):806–815.

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(1):1–9.

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Gabriel Murray. 2019. Discourse analysis and its applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–17.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dee Lang. 2007. Consultant report-natural language processing in the health care industry. *Cincinnati Children's Hospital Medical Center, Winter*, 6.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8180–8187.

Ying Li, Sharon Lipsky Gorman, and Noémie Elhadad. 2010. Section classification in clinical notes using supervised hidden markov model. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 744–750.

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. *arXiv preprint arXiv:1802.05695*.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Simone Teufel, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 110–117.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for icd coding from clinical text. *arXiv preprint arXiv:2007.06351*.

Pengtao Xie and Eric Xing. 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)*, pages 1066–1076.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658.

Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic icd coding via interactive shared representation networks with self-distillation mechanism. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5948–5957.