IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 26, 2020, accepted August 3, 2020, date of publication August 17, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017382

A Hybrid BERT Model That Incorporates Label Semantics via Adjustive Attention for Multi-Label Text Classification

LINKUN CAI, YU SONG, TAO LIU, AND KUNLI ZHANG
School of Information and Engineering, Zhengzhou University, Zhengzhou 450001, China

Corresponding author: Kunli Zhang (ieklzhang@zzu.edu.cn)

This work was supported in part by the Major Program of the National Social Science Foundation of China under Grant 18ZDA315, in part by the Science and Technique Program of Henan Province under Grant 172102410065 and Grant 192102210260, in part by the Medical Science and Technique Program Co-Sponsored by Henan Province and Ministry under Grant SB201901021, and in part by the Key Scientific Research Program of Higher Education of Henan Province under Grant 19A520003 and Grant 20A52003.

ABSTRACT The multi-label text classification task aims to tag a document with a series of labels. Previous studies usually treated labels as symbols without semantics and ignored the relation among labels, which caused information loss. In this paper, we show that explicitly modeling label semantics can improve multi-label text classification. We propose a hybrid neural network model to simultaneously take advantage of both label semantics and fine-grained text information. Specifically, we utilize the pre-trained BERT model to compute context-aware representation of documents. Furthermore, we incorporate the label semantics in two stages. First, a novel label graph construction approach is proposed to capture the label structures and correlations. Second, we propose a neoteric attention mechanism—adjustive attention to establish the semantic connections between labels and words and to obtain the label-specific word representation. The hybrid representation that combines context-aware feature and label-special word feature is fed into a document encoder to classify. Experimental results on two publicly available datasets show that our model is superior to other state-of-the-art classification methods.

INDEX TERMS Multi-label text classification, label embedding, BERT, attention mechanism.

I. INTRODUCTION

Multi-label text classification (MLTC) is a fundamental and challenging task in natural language processing. The purpose of MLTC is to assign a given text with multiple labels. MLTC has been widely applied in many fields such as sentiment analysis [1], intent recognition [2], and recommendation system [3]. With the development of deep learning, single-label classification has made a great success [4], [5], [6]. By treating the problems as a series of single-label classification tasks, the single-label text classification can be naively extended to MLTC task [7]. However, such oversimplified extensions often bring poor performance. Unlike conventional single-label classification (where labels are independent), there are semantic dependencies among various labels. Besides, label relationships can provide implicit and supplemental information, especially when some labels do not have enough training examples. For example, academic literature tagged with ''artificial intelligence'' is usually accompanied by ''deep learning''. In addition, in single-label classification, the prior knowledge of mutual exclusion between labels has been shown and modeled in the final classification. The relationships between multi-label classification are more complicated and not implicitly modeled. Exploiting label correlations has become the primary impetus to improve classification performance [8].

Compared with the shallow model, the deep neural networks have achieved satisfactory performance in the MLTC task [9], [10], [11]. However, they mainly depend on document-level representation. The semantic relationships between labels and word-level information of documents are not modelled explicitly. In other words, the fine-grained document information which will provide clear classification clues is ignored. For example, in information retrieval, related terms such as ''missiles'' and ''tanks'' may be useful to distinguish ''military'' and ''technical'' from various types of documents. Obviously, the words in one document make different contributions to each label.

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Wang.

From the above analysis, MLTC needs to focus on the following two points: 1) how to adequately mine and make use of the correlations among labels. 2) how to extract the discriminatory information of corresponding labels from original documents. Our intuition is to model the labels through a global perspective first and then use the semantic information of labels as the guidance to capture important fine-grained document information.

Recently, a new research direction called graph embedding representation has attracted wide attention [12]. Of all the methods of embedding, the Graph Convolutional Network (GCN) [13], [14] is very beneficial for tasks with a rich relational structure. GCN can retain the global structure information of the graph, thus capturing the semantic dependencies among multiple labels from the perspective of spatial context.

In this paper, we propose a **H**ybrid **B**ERT model incorporates **L**abel semantics via **A**djustive attention (HBLA), which searches and identifies semantic dependencies of label space and text space simultaneously. Firstly, we model the label correlations by label graph built from adjacency-based similarity and then encode the label graph by using GCN, which captures structure information and the rich semantic correlations among labels. Moreover, to capture the label-related discrimination information from each document, we use Bidirectional Encoder Representation from Transformers (BERT) [15] to obtain the implicit representation in the context of each word. An innovative attention mechanism–adjustive attention is proposed to calculate the semantic relationship between word and label explicitly and then based on it to generate label-special word representation. Compared with normal attention methods, the focus area of adjustive attention mechanism becomes more meaningful and discriminatory.

To sum up, we achieve the following contributions in this paper:

- This paper proposes an HBLA related to label graph embedding, which can simultaneously model document and label, and obtains the hybrid word representation.
- We design a novel attention mechanism called adjustive attention to measure the semantic relation between word and label. Adjustive attention learned from word-label is to weight the important fine-grained semantic information in a document.
- Experimental results on two widely-used benchmark datasets achieve superior performance over previous state-of-the-art methods. Extensive validation experiments can prove the effectiveness of label graph embedding and attention mechanism.

The remainder of this paper is organized as follows: Section II introduces related works about the multi-label classification and label embedding methods. Section III describes the HBLA model in detail. Section IV gives extensive experiments to validate the effectiveness of our approach. Experimental results and discussion are given in section V and section VI, respectively. In Section VII, we explore the application of the HBLA in the medical field. The conclusion and future work are summarized in Section VIII.

II. RELATED WORK

A. MULTI-LABEL CLASSIFICATION

The current models for the multi-label classification task can be categorized into three methods: problem transformation, algorithm adaptation, and neural network.

Problem transformation methods are algorithm-independent that transform the multi-label classification task into multiple single-label learning tasks by decomposing the sample set. The most common approach is the Binary Relevance (BR) algorithm, whose core idea is to treat each label as a separate class classification problem and train a binary classifier for each class separately, but without considering label correlations [16]. To fully capture the relationships among labels, the Label Powerset (LP) algorithm treats each combination of labels as a new class. This method is highly complex in training due to the exponential increase of the label [17]. The Classifier Chain (CC) algorithm connects labels in a ''chain'' manner. The prediction of one label can help predict another label to a certain extent. However, these methods only capture low-order correlations [18].

Algorithm adaptation methods, as the name suggests, are to transform the algorithm to deal with multi-label problem. Rank-SVM adopts decision tree based on multi-label entropy for multi-label classification [19]. ML-KNN determines the label set for each sample using the k nearest neighbour algorithm and the maximum posterior principle was used to determine the label set of each sample [20]. RAKEL uses random label subsets as the training set for each LP classifier, and finally integrates predictions of multiple LP classifiers by voting [21].

With the development of deep learning, neural network models have performed well in multi-label classification tasks. BP-MLL algorithm captures the features of multi-label learning by replacing its error function with a pairwise ranking loss function defined [22]. Considering that the labels tend to be correlated, CNN-RNN model combines CNN and RNN to capture local and global semantic information and model high-order correlations among labels with lower complexity [23]. SGM [10] used seq2seq structure to model the relationships between multiple labels and used a gate mechanism to consider global label information. Ashutosh *et al.* explored the fine-tuning BERT for document classification [24].

Different from those approaches that are based on document-level features, we propose the HBLA model with adjustive attention to building label-special word representation, which sufficiently exploits both document content and global label semantics.

B. LABEL EMBEDDING METHODS

The effectiveness of label embedding has been proved in various multi-label learning tasks. The goal of label embedding is to map the label space to low-dimensional vectors and preserve label dependencies.

In computer vision, there is much research on label embedding in image node classification [25] and image

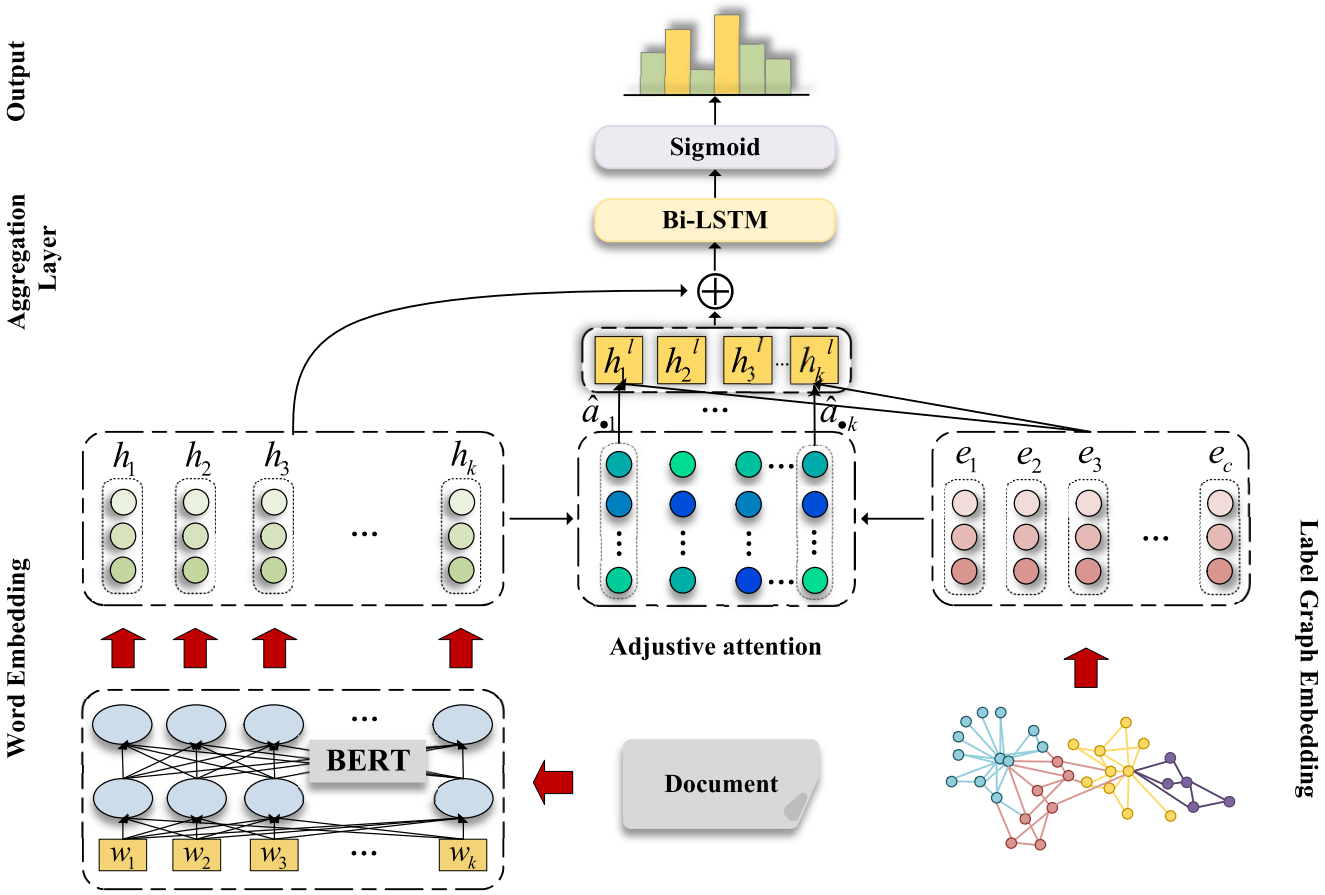

FIGURE 1. The architecture of HBLA model.

recognition [26]. In natural language processing, [27] proposed joint word and label embedding to learn text representation. Reference [28] used label distribution sequences to capture potential long-range label dependencies to improve the performance of sequence labelling.

The graph has been proved to be more effective for label structure modeling. Reference [29] is the first to propose the use of label graph to deal with MLTC, but the shallow neural network model is used for graph representation, and there are limitations in learning the complicated relationships between graph nodes.

Recently, GCN has witnessed prevailing success in modeling relationships among vertices of a graph, a neural network operating on graphs, suited to model syntactic dependency graphs. In this paper, we construct a label graph based on the distribution of labels in the dataset and use GCN to map nodes in the label graph to the same space. Moreover, a novel loss function is designed to constrain nodes in the space. Concretely, labels with more similar distributions are closer in space, otherwise are farther. By separating the non-adjacent nodes, it is possible to capture high-order semantic correlations among labels using network topology structure.

III. MODEL

In this section, we introduce the HBLA model in detail, leveraging the attention mechanism to incorporate label representation and fine-grained word-level representation of documents. As can be seen from Fig. 1, HBLA mainly contains four components:

- Word embedding module projects the input word of a document into context-aware representation.
- Label graph embedding module takes the label graph as input to learn label embeddings which encode the semantic correlations among labels.
- Adjustive attention module calculates the attention scores between word and labels to generate label-specific word representation.
- Aggregation layer integrates the proper information from two aspects (context-aware word representation and label-specific word representation) and uses the hybrid word representation for classification.

A. PROBLEM DEFINITION

Given a dataset $\mathcal{D} = \{D, \mathcal{L}\}$, where $D = \{< d^{(i)}, y^{(i)} >\}_{i=1}^{N}$ means documents and its corresponding targets, and $\mathcal{L} = \{\lambda_1, \cdots, \lambda_c\}$ is a finite set of predefined labels.

The MLTC task can be modeled as learning a function f that maps input document d to binary vectors $\hat{y}$ (assigning a value of 0 or 1 for each label in $\hat{y}$).

B. WORD EMBEDDING

The first part of our model is the word embedding module, which embeds the original words into vectors with

low dimensional. Conventional methods such as Word2Vec [30] and Glove [31] are a kind of fixed word vector representation method. They assume the word with similar meanings no matter whatever contexts it appears. However, the polysemous challenge makes context-independent word embedding difficult for the classification task. For example, the word "apple" would have the same context-independent representation in "APPLE Inc." and "apple juice". To better represent the text content, we compute context-aware representations for each word by using the pre-trained BERT model, which is based on a multi-layer bidirectional Transformer [32] that generates different word embeds for a word in different contexts. BERT takes the input of a sequence of no more than 512 tokens and outputs the representation of the sequence.

Let d be an input document consisting of k words, denoted as $[w_1, w_2, \ldots, w_i, \ldots, w_k]$, where w_i refers to the i^{th} word in the text. A visualization of BERT's architecture is shown in Fig. 1 (bottom left). The arrows indicate the information flow from one layer to the next. The sequence of $H = [h_1, h_2, \ldots, h_k]$ indicates the contextualized representation of each input word.

C. LABEL GRAPH EMBEDDING

We focus on using the label graph to reflect the label structures and represent the label graph in a low-dimension latent space. In this paper, an adjacency-based similarity label graph construction method is proposed to model the interdependencies among labels. We treat each label as a node, and each node gathers features from all neighbours to form its representation. Each edge reflects the semantic correlations between the nodes. If the labels co-exist, there is an edge. This is a flexible way to capture the topology in the label space. The co-occurrence of labels can be described as a joint probability, which is suitable for modeling the labels relationships.

To be more special, we formally define the label graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with C nodes $v_i \in \mathcal{V}$ refers to label λ_i, edges $(v_i, v_j) \in \mathcal{E}$, an adjacency matrix $A \in \mathbb{R}^{C\times C}$ and a degree matrix $D_{ii} = \sum_j A_{ij}$. The adjacency matrix $A = \{A_{ij}\}_{i,j=1}^{c}$ contains non-negative weights between any two nodes. We build this adjacency matrix through a data-driven way. Firstly, we count the occurrence among all label pairs using the label annotations of samples on the training set and get the matrix $\mathcal{C} \in \mathbb{R}^{C\times C}$, by using this label co-occurrence matrix, we can get the adjacency matrix by

$$A_{ij} = log\left(\frac{p_{ij}}{p_{ii} * p_{jj}}\right), \quad i \neq j \tag{1}$$

$$p_{ij} = \frac{\mathcal{C}_{ij}}{\sum_{i,j}^{C} \mathcal{C}_{ij}} \tag{2}$$

$$A = A + I \tag{3}$$

where $\mathcal{C}_{ij}$ means the co-occurrences of label λ_i and λ_j. I is the identity matrix which means that every node is assumed to be connected to itself.

We also construct a word-label adjacency matrix B in the same way as (1) and (2), where B_{ij} is the relationship between w_i and λ_j. $\mathcal{C}_{ij}$ for B means the co-occurrences of w_i and λ_j in samples.

The label embedding is determined from the label co-exist graph and captures the label semantic information defined by the graph structure. We introduce GCN to propagate messages through the graph and learn contextualized label embedding. GCN aggregates the values of all neighboring nodes to update the current node. Each convolutional layer processes only the first-order neighborhood information. Multi-order neighborhood information can be achieved by stacking several convolutional layers. Our goal is to represent the labels in a low-dimensional latent space so that two nearby labels in the graph have similar representation and non-adjacent nodes to be mutually exclusive. For each node $v_i \in \mathcal{V}$, we first initialize with one-hot vector $e_i^{(0)} \in \mathbb{R}^C$. Then, the label embedding can be expressed as:

$$e_i^{(k)} = \rho\left(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \left(\tilde{A}_{ij} \cdot \Theta \cdot e_j^{(k-1)}\right)\right) \tag{4}$$

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{5}$$

where $\tilde{A}$ is the normalized symmetric adjacency matrix and $\Theta \in \mathbb{R}^{C\times C}$ is a trainable weight. $\mathcal{N}(i)$ means the neighbor nodes of i. ρ is a ReLU activation function. In this paper, we consider a two-layer GCN [13], [33] for label embedding which means k is set to 2. Then, we can achieve the label embedding set $E = [e_1, e_2, \ldots, e_c]$.

D. ADJUSTIVE ATTENTION

The attention mechanism is able to capture the global importance of word tokens [34]. As mentioned above, words have fine-grained information for classification, i.e. the word "missiles" is preferable to class "military". Our attention module explicitly introduces the rich label semantics, attentional regions are more meaningful and discriminatory.

However, label space and word space exist a semantic gap. Here we firstly project word space into the label space. We employ a fully connected layer ϕ to re-encoder word representation.

$$H^* = \phi(H) \tag{6}$$

where $H^* \in \mathbb{R}^{K\times C}$. We adopt an attention operator to calculate the attention scores between the target word t and each label. A simple way is calculating the dot-product between H_t^* and E, of which the formulation is:

$$I_t = H_t^* E^T \tag{7}$$

The attention $I_t \in \mathbb{R}^C$ is normalized by $softmax$.

$$a_t = softmax(I_t) \tag{8}$$

For those documents that relate to few labels, the other labels can be regarded as redundant information there and in which case, filtering out unnecessary information plays a relatively essential role.

In order to focus on fine-grained classification clues so that mitigating the irrelevance and redundancy of document contents, we propose the adjustive attention in this paper based on the dot-product attention mechanism. The model dynamically assigns the weight of the label to the word through adjustive attention.

As the degree of association between a word token and the class label may impact their attention scores, the adjustive attention can be divided into two stages. The task of the first stage is to judge the correlation between word and label, we regard this task as a binary classification task, therefore the *sigmoid* function is adapted. If some of the correlation scores are less than threshold τ, we consider that the word is irrelevant to those labels.

In the second stage, the attention score is calculated by *softmax* as above which normalizes the probability distribution. Therefore, the weight of irrelevant labels is reduced, and the weight of relevant labels is enlarged.

The overall operation is described as the following equations:

$$a_t = sigmoid(I_t) \tag{9}$$

$$\hat{a}_{ti} = \begin{cases} 0, & a_{ti} < 0.5 \\ a_{ti}, & else \end{cases} \quad i \in [1, c] \tag{10}$$

$$\hat{a}_t = softmax(\hat{a}_t) \tag{11}$$

Then, the adjustive attention is used to weightily average the label embedding for the word t.

$$h_t^l = \hat{a}_t E \tag{12}$$

where $h_t^l \in \mathbb{R}^C$ is the label-specific word representation, it gives a thought that different labels have inherent characteristics that can be distinguished. Finally, label-special word sequence can be represented via $H^l = [h_1^l, h_2^l, \ldots, h_k^l]$.

The label graph embedding module encodes a label graph to label embedding. The combination of the attention module and label graph embedding module can be regarded as processes of clustering and aggregating. The purpose is to learn a prototype representation for each class and then based on it to generate the label-specific word representation, which aggregates the label semantic.

E. AGGREGATION LAYER

After the above steps, we can obtain two kinds of word representations H and H^l. The former cares about the meaning of words in context, while the latter focuses on the semantic relation between word and label. This layer is designed to aggregate the information from two aspects. For simplicity, the embedding H and H^l are merged by concatenation as shown in (13), where $\hat{H} \in \mathbb{R}^{C+D}$ is the final hybrid word embedding and then provided as input to the document encoder.

$$\hat{H} = H \oplus H^l \tag{13}$$

Then, we use a bidirectional long short-term memory network (Bi-LSTM) [35] as the document encoder to generate document representation. The Bi-LSTM can learn the word embedding for each input text through the forward and backward side. At the time t, the hidden state can be formulated as:

$$\begin{aligned} \overrightarrow{h}_t &= \overrightarrow{LSTM}(\overrightarrow{h}_{t-1}, \hat{h}_t), \quad t \in [1, k] \\ \overleftarrow{h}_t &= \overleftarrow{LSTM}(\overleftarrow{h}_{t-1}, \hat{h}_t) \\ h_t &= \overrightarrow{h}_t \oplus \overleftarrow{h}_t \end{aligned} \tag{14}$$

We use the final hidden state h_k to represent the whole document. Finally, we input h_k to a classifier to predict the confidence score of each label for the document. The classifier consists of a fully connected layer and a *sigmoid* function:

$$\hat{y} = sigmoid(Wh_k^T) \tag{15}$$

here $W \in \mathbb{R}^{C \times (C+D)}$ is the trainable parameter of fully connected layer. D is the word vector dimensions.

F. LOSS

Similar to previous studies [10], we use binary classification loss as our loss function for the MLTC task:

$$\mathcal{L}_c = -\frac{1}{N} \sum_{i=1}^{N} (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \tag{16}$$

Besides, we restrict the label graph embedding so that similar labels are closer together in the label semantic space and non-adjacent labels to be mutually exclusive. One way to encode such property is to make the cosine similarity $\Phi(e_i, e_j)$ to be close to the corresponding edge weight A_{ij} for all i, j. The loss of label graph embedding can be formulated as:

$$\mathcal{L}_g = \sum_{i=1}^{C} \sum_{j=1}^{C} (\Phi(e_i, e_j) + A_{ij} - 1)^2 \tag{17}$$

As mentioned above, we regard the label embedding module and attention module as a clustering process, which requires the label-special word representation to be closer to the centre of its category. Hence, we have designed another loss function to measure the result of clustering, which can be formulated as:

$$\mathcal{L}_e = \sum_{i=1}^{K} \sum_{j=1}^{C} (\Phi(h_i^l, e_j) + B_{ij} - 1)^2 \tag{18}$$

Finally, we define our loss function as follows:

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_c + \mathcal{L}_e \tag{19}$$

IV. EXPERIMENTS

In this section, we evaluate the proposed model on two standard benchmark datasets AAPD [10] and RCV1-V2 [36] to verify the performance.

A. DATASETS

1) ARXIV ACADEMIC PAPER DATASET (AAPD)

The AAPD dataset is a large dataset for MLTC. It consists of 55,840 abstracts of papers about computer science from Arxiv.[1] An academic paper may have multiple subjects, and there are 54 subjects in total.

[1] https://arxiv.org/

TABLE 1. The statistics of AAPD & RCV1-V2.

Datasets	Total	Train	Dev	Test	Labels
AAPD	55,840	53,840	1000	1000	54
RCV1-V2	804,414	802,414	1000	1000	103

2) REUTERS CORPUS VOLUME I (RCV1-V2)

This dataset consists of over 800 K manually annotated news made available by Reuters Ltd for research purposes. Multiple topics can be assigned to each news, and there are 103 topics in total.

Table 1 shows the descriptive statistics of datasets used in our experiments.

B. DETAILS

For pre-processing details, we use WordPiece tokenizer[2] to tokenize the text and lowercase all characters. Each text is limited to 510 tokens.

For model details, our models are implemented by deep learning framework Pytorch Geometric (PyG) [37] and Allennlp [38] and trained on a single GTX1080TI GPU. The label embedding is initialized with one-hot vector. The hidden size of LSTM is 100. We use Adam [39] as the optimizer with the initial learning rate of 2e-5, and the batch size is set to 8. We set the dropout as 0.3 to prevent overfitting and clip the gradients to the maximum norm of 5. Other parameters in our model are initialized randomly.

C. EVALUATION METRICS

To fairly compare the results of our model with those baselines, we adopt the Hamming Loss, Micro-Precision, Micro-Recall and Micro-F1 score as our main evaluation metrics, which are defined as below.

- **Hamming-Loss** is used to measure the mismatches between the real and predicted labels, which is calculated as follows:

$$HL = \frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C} 1(y_{i,c} \neq z_{i,c}) \tag{20}$$

  where $y_{i,c}$ is the target and $z_{i,c}$ is the prediction. C is the numbers of labels and N is the sample size. 1 is an indicator function.

- **Micro-Precision** is the fraction of relevant instances among the retrieved instances, while **Micro-Recall** is the fraction of the total amount of relevant instances that were actually retrieved. And **Micro-F1 score** will be simply the harmonic mean of Precision and Recall. The calculation formulas of the three metrics are as follows:

$$Precision = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FPs(c_i)} \tag{21}$$

$$Recall = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FNs(c_i)} \tag{22}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{23}$$

  where TPs, FNs and FPs represent the number of true positives, false negatives and false positives, respectively.

[2]https://github.com/huggingface/transformers/blob/master/src/transformers/tokenization_bert.py

D. BASELINES

To prove the effectiveness of our proposed model, we use the following baselines to compare it on two aforementioned datasets:

- **BR** decomposes multi-label classification task to multiple independent binary classification problems [16].
- **CC** considers the previous predicted label value and links binary classifiers in a given sequential order [18].
- **LP** combines any two labels in the label set as a new category and transforms a multi-label problem into a multi-class problem [17].
- **CNN** uses a convolution network to extract the text features, then input the features into a fully connected layer. Finally, the sigmoid function is used to output the probability distribution in the label space [40].
- **Seq2Seq Attn** adapts the deep Seq2Seq model with attention mechanism to perform multi-label classification [41].
- **SGM** views the multi-label classification task as a sequence generation problem, and applies a novel decoder structure and an attention mechanism to solve it [10].

V. RESULTS

The quantitative results of our model and all the baselines are shown in Table 2 and Table 3. In each line, the best result is boldface. Results show that the HBLA model outperforms most of the existing methods and reaches a new state-of-the-art performance in the main evaluation metrics. Take the experimental result on AAPD dataset as an example, and some points are observed as follows:

- The models based on deep learning achieve better performance than conventional models on most metrics. Conventional methods are inadequate to cope with more and more complex data, so novel deep learning approaches are becoming increasingly popular. They can take full advantage of the supervision from the training set and can capture more precious features and more in-depth semantic information of texts.
- Compared with CNN which only considers the document content, HBLA reduces the Hamming Loss by 12.9%, and improves the Micro-F1 score by 12.1%. It shows that modeling label semantic correlations can bring performance improvement.
- We can conclude that models with attention outperform other baselines by a large margin. The adjustive attention we propose can incorporate vital fine-grained classification clues to generate label-special word representation. Compared with other attention models, HBLA achieves a Hamming Loss reduction by 5.5% to 14.6%. We detailedly discuss the effect of attention mechanism below.

TABLE 2. Performance on the AAPD test set.

Model	Hamming Loss	Micro-Precision	Micro-Recall	Micro-F1
BR	0.0316	0.664	0.648	0.646
CC	0.0306	0.657	0.651	0.654
LP	0.0312	0.662	0.608	0.634
CNN	0.0256	**0.849**	0.545	0.664
Seq2Seq Attn	0.0261	0.720	0.639	0.677
SGM	0.0245	0.748	0.675	0.715
HBLA	**0.0223**	0.768	**0.722**	**0.744**

TABLE 3. Performance on the RCV1-V2 test set.

Model	Hamming Loss	Micro-Precision	Micro-Recall	Micro-F1
BR	0.0086	0.904	0.816	0.858
CC	0.0087	0.887	0.828	0.857
LP	0.0087	0.896	0.824	0.858
CNN	0.0089	**0.922**	0.798	0.855
Seq2Seq Attn	0.0081	0.889	0.848	0.868
SGM	0.0075	0.897	0.860	0.878
HBLA	**0.0063**	0.906	**0.892**	**0.899**

- In particular, traditional CNN is prior to all current baseline models (including ours) on Micro-Precision evaluation. The main reason is that both datasets we used are imbalanced. CNN is very suitable for extracting local features, and after max-pooling, it would magnify such features, which makes CNN-based classification usually rely on the most obvious features. When the positive examples are more than the negative examples, CNN is more inclined to generate features favourable to the positive examples, which leads to the prediction result skews toward the positive samples while ignoring other categories in the CNN classification. We can also see that Micro-Recall is relatively low. In addition, CNN has little hyper-parameter tuning and static vectors on the classification tasks may be another reason.

Table 3 presents the results of the HBLA and the baselines on the RCV1-V2 test set. It is clear to see that the proposed model is superior to other baseline models in the main evaluation metrics. Different from these methods, HBLA incorporates label semantics to learn better label-specific feature representations, leading to notable performance improvement on all metrics.

VI. ANALYSIS AND DISCUSSION

HBLA contains two critical modules that work cooperatively, i.e., the label graph embedding module and attention module. It is necessary to demonstrate the contributions of each part for the MLTC task. In this section, we conduct some further explorations and analyses of the proposed model on the AAPD dataset.

A. EFFECT OF LABEL GRAPH EMBEDDING

To validate the contribution of the label graph embedding module, we compare our model with the following three models:

1) The BERT model. We chose the pre-trained uncased BERT_{base} model for fine-tuning.
2) Remove the attention module and the label graph embedding module (namely **HBLA-A**). This model can be seen as a combination of BERT and BiLSTM. We only use the semantic features of the document to classify, without considering the effectiveness of the labels.
3) Remove the GCN layers in the label graph embedding module (namely **HBLA-B**). We represent semantic features for each label with one-hot vector, which is a common category encoder that assumes labels are independent of each other.

To avoid the interference caused by word embedding, we use the pre-trained Glove word vector to encode words of documents (words that do not appear in Glove are initialized randomly following a uniform distribution). Similar to the process of the above settings, we remove related modules respectively in two ways (namely **Glove-A**, **Glove-B**). The **Glove** model means only using the Glove word vector to encode documents. The rest modules are unchanged of HBLA.

As shown in Table 4, we notice the gap between the HBLA-family and the Glove-family model. Compared with Glove-A, HBLA-A achieves a Hamming Loss reduction of 13.3% and a Micro-F1 score improvement of 11.2%. HBLA-B and HBLA have a similar trend, which indicates that BERT could better represent the word semantics than Glove, and thus it is easier to capture the correlations between words and labels.

Although simple one-hot vector is applied to represent label feature, the type B models (Glove-B and HBLA-B) perform better than type A models (Glove-A and HBLA-A) in main evaluation metrics. That is to say, the one-hot vector can

TABLE 4. Performance on the AAPD test set with different label embedding methods.

Model	Hamming Loss	Micro-Precision	Micro-Recall	Micro-F1
Glove-A	0.0264	0.7601	0.5805	0.6582
Glove-B	0.0261	0.7609	0.6151	0.6760
Glove	**0.0257**	**0.7632**	**0.6231**	**0.6861**
BERT	0.0231	0.7649	0.7016	0.7320
HBLA-A	0.0229	0.7655	0.7013	0.7320
HBLA-B	0.0227	0.7662	0.7039	0.7337
HBLA	**0.0223**	**0.7684**	**0.7215**	**0.7442**

TABLE 5. Performance on the AAPD test set with different attention mechanism.

Model	Hamming Loss	Micro-Precision	Micro-Recall	Micro-F1
$\text{HBLA}_{average}$	0.0237	0.7501	0.7152	0.7288
$\text{HBLA}_{dot-attn}$	0.0229	0.7567	0.7193	0.7375
HBLA	**0.0223**	**0.7684**	**0.7215**	**0.7442**

partly represent label features, even if they are independent of each other. Meanwhile, the attention module can also aggregate such label features to generate label-specific word representation.

These observations can further validate the label embedding and attention mechanism which are central roles for MLTC task. Both type A models only focus on the representation of the document and ignore the effectiveness of labels on classification. In addition, adding label embedding yields the evident gain over basic BERT, providing significant advance results for classification in this dataset. Our label graph embedding module ensures that HBLA could capture the correlations among labels, including label structure and semantic information.

B. EFFECT OF ATTENTION MECHANISM

We conduct two types of attention experiments to discover the effect of the attention mechanism. First, we design the $\textbf{HBLA}_{average}$ model without attention mechanism (average the corresponding semantic vectors), to explore whether it is necessary to establish a relationship between the label and word-level information of a document. Then, to further verify the effectiveness of adjustive attention, we replace it in the HBLA model with the normal "dot-product attention" and name the replaced model as $\textbf{HBLA}_{dot-attn}$.

Table 5 reports the results under various attention mechanisms. $\textbf{HBLA}_{dot-attn}$ and HBLA models signify that extracting the semantic relation between each word and label is essential and profoundly meaningful. Besides, adjustive attention of HBLA can focus on fine-grained text information and further improve the performance. By contrast, the $\textbf{HBLA}_{average}$ model directly averages the label embedding instead of aggregating different label information for words. That causes the attention module and label embedding module to provide noise for the model, but relies on powerful pre-trained BERT model, $\textbf{HBLA}_{average}$ also achieves a passable result.

To better demonstrate that the adjustive attention can allocate the weight of attention reasonably of different labels,

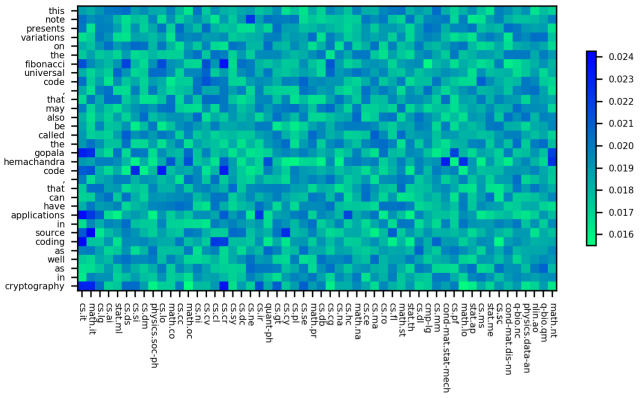

FIGURE 2. Dot-product Attention Visualization.

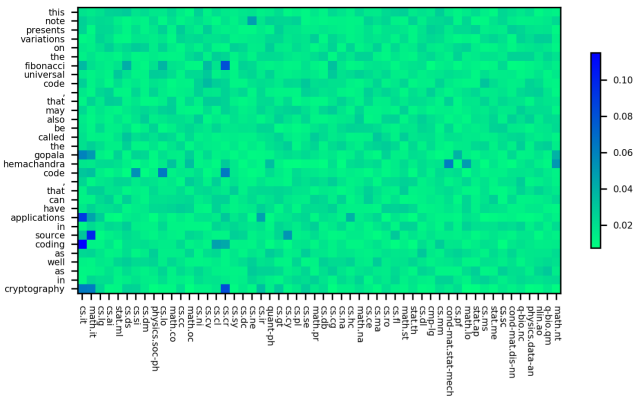

FIGURE 3. Adjustive Attention Visualization.

we pick an example from the AAPD dataset and visualize the attention. As shown in the Fig. 2 and Fig. 3, the attention weights of the correlative labels against words are illustrated. Darker colour refers to a higher weight on the keywords. Compared to the dot-product attention, we can observe that the adjustive attention model can reduce the influence of other label items on the weight distribution of current words and enhance the weights between label items corresponding to current words. In this case, the word *coding*, *applications* and *cryptography* support tagging this document with *cs.it*.

TABLE 6. Performance on the MIMIC3 test set.

Model	Macro-AUC	Micro-AUC	Macro-F1	Micro-F1
Logistic Regression	0.829	0.864	0.477	0.533
CNN	0.876	0.907	**0.576**	**0.625**
Bi-GRU	0.828	0.868	0.484	0.549
Attentive LSTM	-	0.900	-	0.532
CAML	0.875	0.909	0.532	0.614
HBLA	**0.897**	**0.922**	0.541	0.620

VII. APPLICATIONS IN THE MEDICAL FIELD

The novel Coronavirus Disease 2019 (COVID-19) presents an urgent threat to global health [42]. The application of deep learning technology in the medical field has been an essential trend to mitigate the burden on the healthcare system. To demonstrate the practical value of our model, we apply HBLA for a real health care scenario: predicting ICD-9 codes(or it could be disease diagnosis) according to patient's electronic medical records (EMRs). Since an EMR usually contains multiple types of disease codes, we treat predicting diagnostic codes as an MLTC task.

We evaluate HBLA on the publicly available MIMIC3 dataset [43], which contains 58,976 ICU EMRs. Each EMR includes clinic text of one patient, and we only focus on discharge summaries, which record information about a stay.

To compare with previous works [44], [45], we use the top 50 codes for experiments which results in 8,066 EMRs for training and 1,729 for testing. The baseline methods include Logistic Regression, CNN, Bi-GRU, Attentive LSTM [44] and Convolutional Attention (CAML) [45]. The last two baseline methods specialize in ICD-9 code prediction task. We apply the F1 score and AUC (area under the ROC curve) to validate performance.

The results are shown in Table 6, and the Logistic Regression baseline performs worse than all deep learning methods. HBLA provides excellent performance on most metrics, which proves that our model is universal and practical. However, since the average length of the discharge diagnosis free text is more than 2,000 tokens, while the word embedding module of HBLA is limited the sequence lengths up to 512 tokens, some information will be lost due to interception. At the same time, due to the particularity of the medical field, the effect of pre-training model is not satisfactory. If biomedical domain corpus is used to learn word representations,[3] they will serve better for classification of domain-specific text documents.

VIII. CONCLUSION

The application scenarios of multi-label classification are very broad, which is a hot topic in the field of natural language processing. Our model introduces pre-trained BERT to obtain word context as well as in-depth semantic information. Moreover, capturing label semantics takes a crucial position in MLTC. To better model that information, we propose an adjacency-based similarity method to construct the label graph and obtain the semantic embedding of the label by using GCN. We propose a novel adjustive attention mechanism to explicitly calculate the semantic relationship between labels and documents to capture useful label-specific information and suppress noise. The final hybrid word representation is used for classification. We also explore the effect of label embedding and adjustive attention. Experimental results on two multi-label classification datasets demonstrate the superiority of HBLA.

Actually, imbalanced data is ubiquitous in the real world, and it may deteriorate the performance of conventional classification algorithms. Imbalanced multi-label data often exhibits skewed distribution compared with single label data. We will verify the improvement of the model on a single label and investigate the performance of the model on some imbalanced datasets in future work.

[3]http://evexdb.org/pmresources/vec-space-models

REFERENCES

[1] B. Myagmar, J. Li, and S. Kimura, "Cross-domain sentiment classification with bidirectional contextualized transformer language models," *IEEE Access*, vol. 7, pp. 163219–163230, 2019.

[2] Y. Papanikolaou, D. Dimitriadis, G. Tsoumakas, M. Laliotis, N. Markantonatos, and I. P. Vlahavas, "Ensemble approaches for large-scale multi-label classification and question answering in biomedicine," in *Proc. CLEF*, 2014, pp. 1348–1360.

[3] L. Guo, B. Jin, R. Yu, C. Yao, C. Sun, and D. Huang, "Multi-label classification methods for green computing and application for mobile medical recommendations," *IEEE Access*, vol. 4, pp. 3201–3209, 2016.

[4] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 1107–1116.

[5] B. Wang, "Disconnected recurrent neural networks for text categorization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2311–2320.

[6] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. 33th AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 7370–7377.

[7] O. Reyes, C. Morell, and S. Ventura, "Scalable extensions of the reliefF algorithm for weighting and selecting features on the multi-label learning context," *Neurocomputing*, vol. 161, pp. 168–182, Aug. 2015.

[8] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[9] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 427–431.

[10] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. Int. Conf. Comput. Linguistics*, 2018, pp. 3915–3926.

[11] P. Yang, F. Luo, S. Ma, J. Lin, and X. Sun, "A deep reinforced sequence-to-set model for multi-label classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5252–5258.

[12] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907
[14] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018, *arXiv:1812.08434*. [Online]. Available: http://arxiv.org/abs/1812.08434
[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[16] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.
[17] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.
[18] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.
[19] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 681–687.
[20] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.
[21] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-Labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.
[22] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
[23] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2377–2383.
[24] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for document classification," 2019, *arXiv:1904.08398*. [Online]. Available: http://arxiv.org/abs/1904.08398
[25] K. Gao, J. Zhang, and C. Zhou, "Semi-supervised graph embedding for multi-label graph node classification," in *Proc. Int. Conf. Web Infor. Syst. Eng.*, 2019, pp. 555–567.
[26] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5177–5186.
[27] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proc. 56th Annu. Meeting Assoc. for Comput. Linguistics*, 2018, pp. 2321–2331.
[28] L. Cui and Y. Zhang, "Hierarchically-refined label attention network for sequence labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 4106–4119.
[29] W. Zhang, J. Yan, X. Wang, and H. Zha, "Deep extreme multi-label learning," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2018, pp. 100–107.
[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781
[31] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.(NIPS)*, 2017, pp. 5998–6008.
[33] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32th AAAI Conf. Artif. Intell.*, 2018, pp. 1–7.
[34] X. Sun and W. Lu, "Understanding attention for text classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3418–3428.
[35] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling," 2016, *arXiv:1611.06639*. [Online]. Available: http://arxiv.org/abs/1611.06639
[36] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004.
[37] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch geometric," in *Proc. ICLR Workshop Represent. Learn. Graphs Manifolds*, 2019, pp. 1–9.
[38] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "AllenNLP: A deep semantic natural language processing platform," 2018, *arXiv:1803.07640*. [Online]. Available: http://arxiv.org/abs/1803.07640
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[40] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.
[41] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 28st Conf. Neural Inf. Process. Syst.(NIPS)*, 2014, pp. 3104–3112.
[42] L. Wynants, "Prediction models for diagnosis and prognosis of covid-19 infection: Systematic review and critical appraisal," *BMJ*, vol. 369, p. 7, Apr. 2020.
[43] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W.-H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, no. 1, Dec. 2016, Art. no. 160035.
[44] H. Shi, P. Xie, Z. Hu, M. Zhang, and E. P. Xing, "Towards automated ICD coding using deep learning," 2017, *arXiv:1711.04075*. [Online]. Available: http://arxiv.org/abs/1711.04075
[45] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2018, pp. 1101–1111.

LINKUN CAI was born in 1996. She is currently pursuing the master's degree in software engineering with Zhengzhou University, China. Her research interests include artificial intelligence and natural language processing.

YU SONG was born in 1969. He received the M.S. degree in applied mathematics from the Huazhong University of Science and Technology, Wuhan, China, in 2001.

He is currently an Associate Professor and a Ph.D. Supervisor with Zhengzhou University. His main research interests include artificial intelligence and machine learning.

TAO LIU was born in 1996. He is currently pursuing the master's degree in computer technology with Zhengzhou University, China. His research interests include artificial intelligence and natural language processing.

KUNLI ZHANG was born in 1977. She received the Ph.D. degree in software engineering from Zhengzhou University, Zhengzhou, China, in 2019.

She is currently a Lecturer with Zhengzhou University. Her main research interests include artificial intelligence and natural language processing.