Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1043872, 12 pages
https://doi.org/10.1155/2021/1043872

Research Article

An Approach Based on Multilevel Convolution for Sentence-Level Element Extraction of Legal Text

Zhe Chen, Hongli Zhang, Lin Ye, and Shang Li

School of Cyberspace Science, Harbin Institute of Technology, Harbin, China

Correspondence should be addressed to Shang Li; ls@hit.edu.cn

Received 15 June 2021; Revised 11 October 2021; Accepted 24 November 2021; Published 24 December 2021

Academic Editor: Peng Li

In the judicial field, with the increase of legal text data, the extraction of legal text elements plays a more and more important role. In this paper, we propose a sentence-level model of legal text element extraction based on the structure of multilabel text classification. Our proposed model contains an encoder and an improved decoder. The encoder applies multilevel convolutional neural networks (CNN) and Long Short-Term Memory (LSTM) as feature extraction networks to extract local neighborhood and context information from legal text, and a decoder applies LSTM with multiattention and full connection layer with an improved initialization method to decode and generate label sequences. To our best knowledge, it is one of the first attempts to apply a multilabel classification algorithm for element extraction of legal text. In order to verify the effectiveness of our model, we conduct experiments not only on three real legal text datasets but also on a general multilabel text classification dataset.The experimental results demonstrate that our proposed model outperforms baseline models on legal text datasets, and our model is competitive to baseline models on the general text multilabel classification dataset, which indicates that our proposed model is useful for multilabel classification tasks of ordinary texts and legal texts with an uncertain number of characters in words and short lengths.

1. Introduction

With the development of the economy, there are more and more civil legal disputes, so that legal practitioners have to deal with more and more legal texts; however, the number of legal practitioners has not expanded with the increase in the number of documents. To alleviate the contradiction between the large number of cases and the small number of legal practitioners in China's judicial field in recent years and to improve the work efficiency of legal practitioners, it is necessary to use automated extraction technology to extract text sentence elements from legal texts to help legal practitioners understand important information in texts quickly. The development of natural language processing (NLP) and the availability of legal texts provide a foundation for the achievement of the above demand. At present, there are relatively few researches about the element extraction of legal texts. In this paper, the extraction of legal text elements is defined as the assignment of labels with specific legal attributes to each sentence in the legal text according to the semantic information it represents. For example, in divorce cases, labels can be "婚后有子女(children after marriage)," "有夫妻共同债务 (joint debt of husband and wife)," etc. In labor cases, labels can be "签订劳动合同 (signed labor contract)," "支付经济补偿金 (pay economic compensation)," etc. In loan cases, labels can be "免除保证人保证责任 (exempt the guarantor from guarantee responsibility)," "保证人不承担保证责任 (guarantor does not assume responsibility for guarantee)," etc. Through the statistical analysis of the labels of legal text datasets, it can be found that there is collinearity among the labels of legal texts; that is, a sample may belong to 0 to N categories at the same time. Therefore, the element extraction of legal text can be regarded as the multilabel classification (MLC) problem of texts rather than a multiclass classification (MCC) problem.

The early method to solve the task of multilabel text classification is to transform it into a number of dichotomy problems [1] and determine whether the sample belongs to each class by setting a threshold value. This approach ignores the correlation between labels which has limited

performance. After that, [2] proposes Classifier Chain (CC) method that applies several binary classifiers to construct a transfer chain for the multilabel text classification task to model the correlation between labels. The performance of this approach is affected by the random arrangement of label categories in the chain and the possibility that the previous classifier in the chain may propagate false predictions along the chain to the next classifier. Decision tree, SVM, and KNN methods are also applied to solve MLC tasks, respectively, in [3–5] which can only capture labels with first or second order correlation and are computationally intractable when high order correlations are required. The multilabel classification for legal text needs to recognize all labels rather than top-K labels, so [3–5] can not be applied to the legal text. With the development of deep neural network technology, many methods based on neural network have been proposed to solve MLC tasks. [6] proposes the fully connection neural network method with a pair-sorted loss function, but this method ignores the label correlations. [7] is the first to address the problem of translating MLC task into sequence-to-sequence (Seq2Seq) text label predictions for the given text which models the correlation between labels. From then on, methods based on the Seq2Seq structure have been widely proposed [8–11]. Research shows that local information is effective for text classification [12–16], but these Seq2Seq-based methods based on simple recurrent neural networks (RNN) or simple CNN have very limited ability in capturing local information of text which reduces the effectiveness of model. The difference between legal text and general text lies in the uncertainty of the number of characters in text words. The number of characters in a word of legal text is not fixed. Therefore, CNN based on a single convolution kernel can only extract the surrounding features of characters in the fixed window size, whose ability to extract features is limited. [17] introduces a model which applies multilayer dilated convolution to extract semantic-unit information. However, the method based on multilayer dilated convolution loses some important semantic information due to the discontinuity of convolution and affects the acquisition of semantic information.

To address these problems, we propose a multilabel legal element extraction model based on Seq2Seq structure, which is composed of encoder and decoder. The encoder adopts the multilevel convolution neural network (MCNN) to alleviate the number of characters in each word in the legal text is not fixed problems, applying different window size of convolution kernels to extract more features, and applies LSTM to capture long-term context dependencies between texts. The decoder is composed of LSTM, multiattention module, and fully connection layer. The LSTM in decoder is applied to model the association between current state and previous labels. Next, at each time step t, the output of the decoder not only applies the feature information encoded by the encoder LSTM, label distribution information that is encoded by LSTM of decoder, but also applies the local semantic information encoded by MCNN through the attention mechanism. Finally, the decoder applies the fully connect layer whose parameter is initialized by an improved method according to the cooccurrence numbers of the labels according to the statistics of the training dataset to generate the output.

To investigate the performance of our proposed model, experiments are conducted on a generic dataset (RCV1V2) for multilabel text classification task and three legal text datasets. The experimental results demonstrate that our proposed model outperforms baseline models on legal text datasets, and our model is competitive to baseline models on the RCV1V2 dataset in evaluation metrics.

Our contributions in this paper are summarized as follows:

(1) To the best of our knowledge, this is the first study that apply the multilabel classification algorithm for element extraction of legal text

(2) We propose a Seq2Seq model containing multilevel convolution network that is applied to alleviate the number of characters in each word in the legal text is not fixed problems by applying different window size of convolution kernels to extract ong distance features, an improved decoder structure based multi-attention and fully connection layer whose parameter is initialized by an improved method

(3) We conduct experiments on three real-world datasets of Chinese legal text and a general multilabel text classification dataset. The results demonstrate that our model is competitive to baseline models on the Chinese legal text and the general multilabel text classification task

The rest of the paper is organized as follows. In Section 2, we briefly review the related work about multilabel text classification. In Section 3, we introduce the Bi-directional Long Short-Term Memory (BiLSTM) and CNN network. In Section 4, we describe the architecture of our model. In Section 5, the experimental results and analyses are presented. In Section 6, concluding is presented.

2. Related Work

Existing multilabel text classification models can be divided into three categories: problem transformation method which is to transform problem data to use existing algorithms, algorithm adaptive method, which extends a specific algorithm so as to be able to process multilabel data, and deep learning method.

In some methods based on problem transformation, the correlation between labels is considered, while in others, it is not. The simplest nonassociative algorithm does not model the correlation between labels. Instead, the labels in multiple label texts are treated as an independent label, and a common classification algorithm is implemented for each label. The Binary Relevance (BR) [1] models a separate classifier for each label, resulting in correlation of labels being ignored. To model label dependencies, Label Powerset (LP) [18] builds a binary classifier for each label group validated in datasets. Classifier Chains (CC) [2] converts the MLC task

into a binary classification problem chain, taking into account higher-order label dependencies.

Instead of transforming the problem into different subsets of the problem, the algorithm adaptive method is applied for multilabel classification directly. ML-DT [3] applies a decision tree with multilabel entropy algorithm for multilabel classification. Rank-SVM [4] applies the support vector machine (SVM) model with similar learning system algorithm for multilabel classification. ML-KNN [5] applies k-nearest neighbor and the maximum posterior principle algorithm to determine the label sets of each sample. [19] ranks the label by using a pairwise comparison. [20] applies CBM to simplify multilabel classification tasks by converting them into standard binary and multiclass problems to perform classification.

In recent years, with the wide application of deep learning in natural language processing (NLP), the method of multilabel text classification based on deep learning has been proposed continuously. [6] proposes a BP-MLL model by applying a fully connected neural network and a pair of sorting loss functions to perform classification. A better training can be obtained by changing the ordering loss function to the cross entropy loss function in [21]. [12] proposes a model based on neural network initialization method, in which some neurons are applied as specialized neurons to model label correlations. [8] proposes a model that applies convolutional neural network and recurrent neural network simultaneously to capture local and global semantic information and construct correlations between labels. [7] proposes to generate labels sequentially. SGM [9] and SU4MLC [17] methods both use the Seq2Seq model structure: one applies an improved decoder with applied global embeddings, and the other contains additional semantic units obtained form dilated convolution with attention to enhance the presentation of information. [22] proposes a multilabel reasoning model based on iterative reasoning mechanism, which uses a binary classifier for each label and predicts all labels at the same time to achieve the disorder of labels.

3. Preliminaries

3.1. BiLSTM. LSTM network is a special form of RNN network, which can solve the long time dependence problem well by storing the past data in its memory cell and can alleviate the problem of gradient disappearance and explosion in RNN network. The LSTM network consists of three gate structures (input gate, output gate, and forget gate) and a memory unit. LSTM can add and delete letters to memory cells through gate structures. Each node of the LSTM network is calculated as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{1}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right), \tag{2}$$

$$\widehat{c} = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right), \tag{3}$$

$$c_t = f_t * c_{t-1} + i_t * \widehat{c_t}, \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t * \tanh\left(c_t\right), \tag{6}$$

where x_t represents the input embedding at time t; W_i, W_f, W_c, and W_o are weight parameter matrices that the network needs to learn; f_t, o_t, and c_t represent forget gate output, output gate output, and cell output, respectively; σ represents the sigmoid activation function; h_t represents hidden outputs at time t; b_i, b_c, b_f, and b_o represents the biased values of the respective gate structures.

For BiLSTM, the forward LSTM unit and backward LSTM unit are simultaneously calculated by equations (1), (2), (3), (4), (5), and (6), respectively. The output $\overleftarrow{h_t}$ of forward and output $\overrightarrow{h_t}$ of backward are then joined together to form the final output h_t at time t. The calculation process is as follows:

$$h_t = \left[\overleftarrow{h_t}, \overrightarrow{h_t}\right]. \tag{7}$$

Finally, the overall output h of BiLSTM network is $h = (h_1, h_2, \cdots, h_n)$.

3.2. CNN. The difference between CNN and ordinary neural networks lies in that contains a feature extractor composed of convolution layer and pooling layer. In the convolutional layer of convolution neural network, one neuron only connects with some adjacent neurons. CNN usually consists of convolutional layer and pooling layer, which is used to capture local features of text classification.

The core of CNN is the convolutional layer that contains a set of convolution kernels. The convolution computation is performed by using convolution kernel and local windows of input embeddings. The calculation process is as follows:

$$c_i = \delta(W \cdot e_{i+l-1} + b), \tag{8}$$

where W represents the convolution kernel parameter and $W \in R^{i*j}$, where i is the height of convolution kernel W and j is the width of convolution kernel W. b represents bias parameter value, δ is a nonlinear function, e_{i+l-1} represents the input embeddings from i layer to $i+l-1$ layer, and c_i is output of convolution calculation. The convolution window moves with the specified step size to capture the local neighbor feature.

The function of the pooling layer is to sample and compress the convolution results to prevent overfitting. The pooling method is divided into maximum pooling and average pooling. Maximum pooling means to maximize the feature points in the neighborhood, which can retain texture information well. Average pooling means only averaging the feature points in the neighborhood, which can preserve the background features well.

$$C_i = \text{pooling}(c_i). \tag{9}$$

4. Method

In this section, we introduce our proposed methods in detail. We firstly give an overall structure of our proposed method in Section 3.1. Then, we introduce the structure of encoder in Section 3.2. The decoder with multiattention and fully connection layer with special initialization will be introduced in Section 3.3.

4.1. Overall. Firstly, we define some symbols to make the presentation clear that describes the multilabel classification (MLC) task. Given a input sequence $X = (x_1, x_2, \cdots, x_n)$, the multilabel classification task is aimed at predicting the label sets $Y \in L$ corresponding to X, where $L = (l_1, l_2, \cdots, l_m)$ is the label space and the number of labels in Y belongs to 1 to m in samples, n is the length of sequence X, x_i is the i -th word in the sequence, and m is the total number of labels. MLC task can be defined to find the maximum conditional probability $p(Y \mid X)$ of labels.

The model structure we proposed is shown in Figure 1, which includes the encoder and the decoder. In the encoder, multilevel CNN consists of multiple CNNs from lower to higher layers is applied to capture the local representation of text, and CNN networks with higher layers can capture more long-distance information. The LSTM is applied to extract the context representation of the texts. For the decoder to output the label results at each time step, it should not only refer to the previous label state but also process the local representation from the output of the multilevel CNN and the context representation that is from the encoder LSTM in a mixed attention to generate the current label state. Finally, a fully connected layer followed by the softmax layer converts the output of the decoder into the final probability distribution for output.

4.2. Encoder. In this section, we introduce the encoder module in detail, which is applied to capture local semantic information by CNN and context representation by LSTM.

CNN has been widely applied in text classification tasks [17, 23–25], because of its strong ability for local semantic extraction. LSTM has also been widely applied in various sequence-to-sequence tasks recently [26] due to greatly capturing context features between words with gate units. We apply multilevel CNN to capture the local semantic feature and LSTM to capture the context feature among words in legal texts in encoder. We run the above two networks in parallel to extract semantic and contextual information of the words.

4.2.1. Word Embedding. Let $(x_1, x_2, \cdots, x_m)$ represent a sentence with m words, where x_i is the i-th word in sentences. In word embedding layer, we first convert x_i to embedding vector e_i by an embedding lookup table $E \in R^{k*|v|}$, a random initialization embedding matrix that can be continuously optimized during training, where $|v|$ is the size of the vocabulary and k is the dimension of the embedding vector.

4.2.2. Multilevel CNN. In CNN, convolution kernel and maxpooling layer are usually applied to extract the most important local semantic features. However, due to the fact that the location feature information between words is ignored during the pooling of the maxpooling layer, part of semantic information of extracted features is missing [25]. Through the above analysis, we apply the multilayer CNN networks without pooling layer to generate local semantic representation units.

In the CNN feature extraction network, a convolution filter $W \in R^{p*d}$ is applied to extract word features in the sentence $(x_1, x_2, \cdots, x_n)$ by window size of p at each layer by moving from left to right to extract local features of words, where d is the dimension of the input embeddings and p is the size in the convolution kernel.

$$g_i^1 = f\left(W * e_{i:i+p-1} + b\right), \tag{10}$$

where $b \in R$ indicates a bias term and f refers to a nonlinear activation function. Finally, we obtained a feature map of words in sentence.

The MCNN applies multiple above filters by varying window sizes with different convolution kernel and stack the multiple layers to obtain a wide range of local information. The output of C_{i-1} is the input to C_i, where C_i is for the i-th convolution network. Specifically, the calculation is as follows:

$$g_i^l = f\left(W * g_{i:i+p_l}^{l-1}\right), \tag{11}$$

where f refers to a nonlinear activation function, $g^l[i : i + p_l]$ is the i-to-$(i + p_l)$-th column in the l-h convolution network, and p_l represents the size in the l-th convolution kernel. We regard the word embedding layer as the first layer g^1. Similarly, the feature in the l-th layer g^l represents the output of l-th convolution network to capture local information. The structure of nultilevel CNN is shown as Figure 2. The final result of MCNN is calculated as follows:

$$\eta_i = \text{Concat}\left[g_i^1, g_i^2, \cdots, g_i^L\right], \tag{12}$$

where L is the number of kernels.

4.2.3. LSTM. We apply the BiLSTM [27, 28] to encode the text input sequence embedding $X = (x_1, x_2, \cdots, x_n)$ and generate the hidden states $h = (h_1, h_2, \cdots, h_n)$ for n word, where the final hidden states are represented by concatenating $(h_i = [\overleftarrow{h_i}\,; \overrightarrow{h_i}])$ at each time step i, where $\overleftarrow{h_i}$ represents the hidden layer state from left to right and $\overrightarrow{h_i}$ represents the hidden layer state from right to left. The calculation methods of h_i in step i are illustrated below:

$$h_i, s_i = \text{LSTM}_{\text{encoder}}(x_i, h_{i-1}, s_{i-1}), \tag{13}$$

where x_i represents the input embedding of step i, h_{i-1} represents hidden state of step $i - 1$, and s_{i-1} represents cell state of step $i - 1$.

4.3. Decoder. In this subsection, we introduce the decoder module in detail. In order to improve the multilabel classification results, our model applies three steps to generate the label sequences in the decoding process. First, an LSTM is

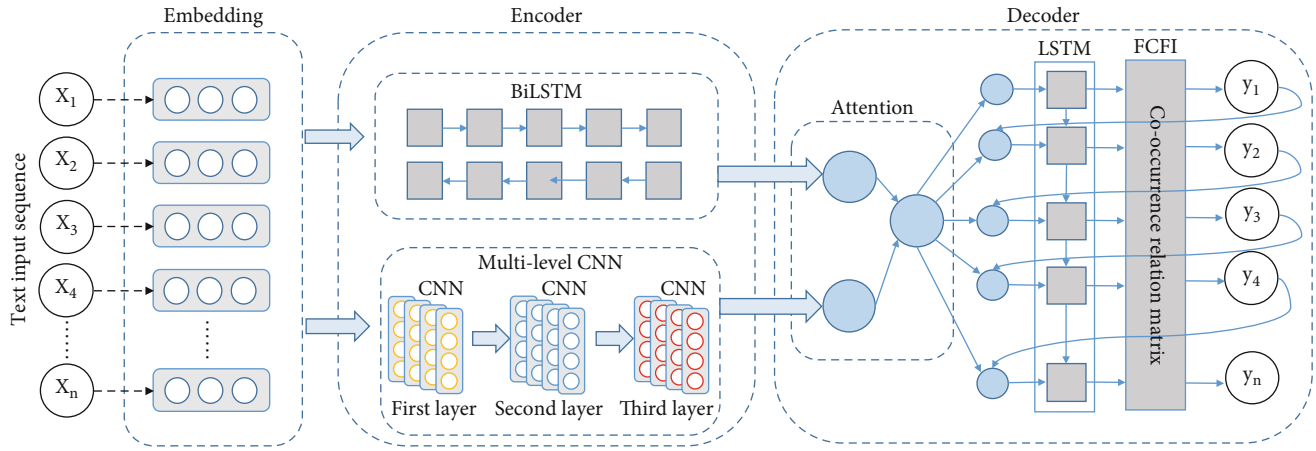

Figure 1: Overall structure of our proposed method.

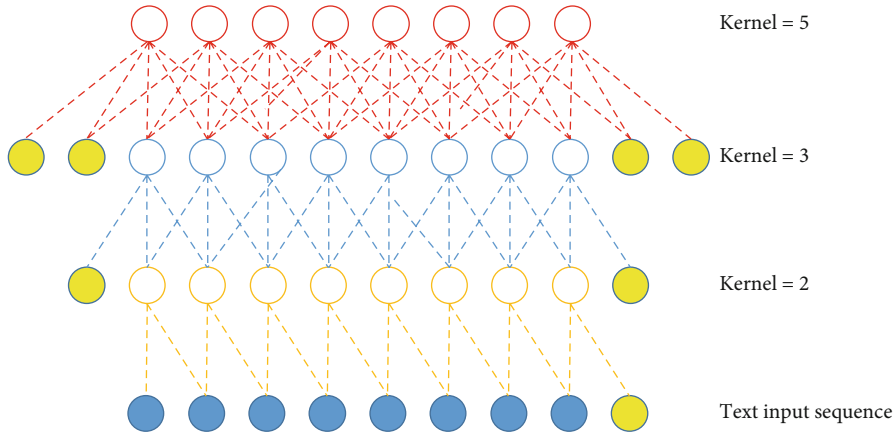

Figure 2: Structure of multilevel CNN. The kernel size in our method is 2, 3, and 5. The multilevel CNN is a multilayer one-dimensional CNN in which number of output channels equal to the number of hidden layer units.

applied to generate the corresponding label state sequence by the relationship between the generated labels. Second, to pay attention to the local semantic information generated from multi-CNN and the context information generated from the LSTM in the encoder, we take advantage of multi-attention to capture feature information inside the sentences and form the final textual representation. Finally, we apply a fully connection layer with improved initialization method as the final output layer.

4.3.1. LSTM. At time-step t, the hidden state s_t^d in LSTM of decoder is computed as follows:

$$\begin{aligned} u_{t-1,i} &= \tanh\left(W_c h_i + W_s s_{t-1}^d\right), \\ r_{t-1,i} &= \frac{\exp\left(u_{t-1,i}\right)}{\sum_{j=1}^{m} \exp\left(u_{t-1,i}\right)}, \\ \iota_{t-1} &= \sum_{i=1}^{m} r_{t-1,i} h_i, \\ a_t &= \text{concat}(y_{t-1}; \iota_{t-1}), \\ s_t^d &= \text{LSTM}_{\text{decoder}}\left(s_{t-1}^d, a_t\right), \end{aligned} \tag{14}$$

where W_c and W_s are weight parameters, h_i is the hidden state in LSTM of encoder at step i, y_t is the output of the method at timestep t, m is the number of words, and concat represents the concatenation operation of the vectors.

4.3.2. Multiattention. Our proposed model learns the semantic features of the text according to the multiattention method in [17]. The structure of multiattention is shown as Figure 3. For the output o_t of the decoder, it not only considers the context features from the LSTM encoding in the encoder but also considers the local semantic features from the multilevel CNN. In our model, the decoder first applies the attention mechanism to pay attention to the local information from the multilevel CNN and decoded sequence information of the labels, calculates the semantic features that can represent the sentence, and generates the new representation. Next, the attention mechanism is applied to pay attention to the newly generated representation, the decoded information of labels, and the text context information captured by LSTM in encoder. Firstly, s_t^d that the LSTM in decoder output and the semantic representations $\eta = (\eta_1, \cdots, \eta_i, \cdots, \eta_m)$ captured by the multilevel CNN in encoder are applied to generate a new representation s_t'

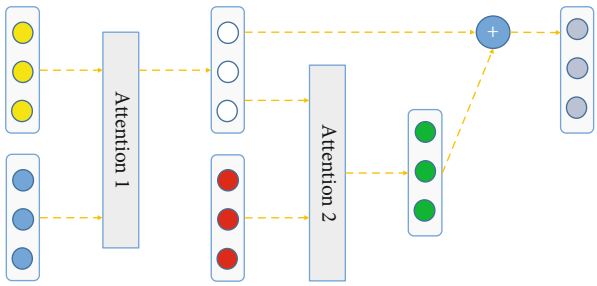

Figure 3: Structure of multiattention. The yellow circles represent output of LSTM in decoder, the blue circles represent the local representations generated by multilevel CNN, the white circles represent s_t', the red circles represent output of LSTM in encoder, the green circles represent $\tilde{s}_t$, and the grey circles represent o_t.

with attention model. Then, the new representation s_t' and context information $h = (h_1, \cdots, h_i, \cdots, h_n)$ captured by the LSTM in encoder are applied to generate another representation $\tilde{s}_t$ with attention model. Finally, s_t' and $\tilde{s}_t$ are added element-wisely to generate the score output o_t of labels at time-step t for the prediction of y_t. The specific is calculated as follows:

$$\begin{aligned}
\varphi_t &= f\left(W_t s_t^d + U_t \eta_t\right),\\
\chi_{ti} &= \frac{\exp\left(\varphi_{ti}\right)}{\sum_{j=1}^{m} \exp\left(\varphi_{tj}\right)},\\
\eta_t' &= \sum_{i=1}^{m} \chi_{ti} \eta_t,\\
s_t' &= W_c\left(\left[\eta_t'; s_t^d\right]\right),\\
e_{ti}' &= W_e\left(W_t s_t + U_t' s_t'\right),\\
\chi_{ti}' &= \frac{\exp\left(e_{ti}'\right)}{\sum_{j=1}^{m} \exp\left(e_{tj}'\right)},\\
\tilde{s}_t &= \sum_{i=1}^{m} \chi_{ti}' h_i,\\
O_t &= s_t' + \tilde{s}_t,
\end{aligned} \tag{15}$$

where W_e, W_c, W_t, U_t, and U_t' are weight parameters, O_t is an m-dimensional vector, and m is the number of labels. Each dimension value represents the probability that the sentence belongs to the corresponding class. The higher the score, the more likely it is to belong to the class.

4.3.3. Fully Connection Layer with Frequency Initialization (FCFI). We propose an initialization method by normalization to better model the cooccurrence between labels. Score output of label o_t' through FCFI layer is calculated as follows:

$$O_t' = W_F O_t, \tag{16}$$

where O_t is output of LSTM in decoder and $W_F \in R^{L*L}$ is weight parameter that is initialized with a symmetry matrix, where L is number of lables. The element in i-th column and j-th row in W_F represents cooccurrence between label i and label j. The initialization value of matrix W_F is calculated as follows:

$$W_{F(i,j)} = \begin{cases} \alpha * \dfrac{\text{count}_{i,j}}{A_j} + \beta * \dfrac{\text{count}_{j,i}}{A_i} & \text{if } i \neq j, \\ 1 & \text{otherwise,} \end{cases} \tag{17}$$

$$A_i = \sum_{j=1}^{L} \text{count}_{i,j}, \tag{18}$$

where α+β=1, $\text{count}_{i,j}$ is the number of cooccurrence between labels i and j in the training dataset, and A_i is the number of samples that contain label i. The initialization value on the diagonal is set to 1. The higher value of $W_{F(i,j)}$ is, the more likely the label i and label j are to appear together.

4.3.4. Softmax Layer. Softmax layer [9] is applied to predict the label y_t at timestep t

$$y_t = \text{softmax}\left(O_t' + \kappa_t\right), \tag{19}$$

where $\kappa_t \in R^L$ is a mask vector that is applied to prevent current label from duplicating the previous label and is calculated as follows [25]:

$$\left(\kappa_t\right) = \begin{cases} -\infty & \text{Previous timesteps } t-1 \text{ has predicted the label,} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

5. Experiments

In the following, we introduce the datasets containing one general multilabel text classification dataset to verify the effectiveness of our model by comparing with models that work well on general datasets and three legal text datasets of civil cases, preprocessing of legal datasets, experimental parameter setup, and related baselines we compare with.

5.1. Datasets

(i) *RCV1-v2* (http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.html). RCV1-v2 [29] is provided for research purposes and consists of more than 800,000 manually categorized news made available by Reuters Itd. Multiple ones can belong to multiple topic types, and the number of topics reached 103

(ii) *Legal Text Datasets*. The legal text dataset is provided by CAIL2019 (https://github.com/china-ai-

law-challenge/CAIL2019). The dataset contains three civil datasets, namely, *Divorce*, *Loan*, and *Labor*. These datasets are from the legal documents published on China Judgements Online (https://wenshu.court.gov.cn/) and marked by experts. Each item of datasets consists of sentences and its corresponding element labels, which are extracted from part of a judgment document

5.2. Preprocessing of Legal Datasets. We preprocess the legal datasets to remove the sentences without any labels and sorted the labels according to the number of times they appear in the dataset from the highest to the lowest. Each subset of legal text dataset has 20 label categories. After statistical analysis of the preprocessed data, there are often correlations between labels. In this paper, the legal text dataset is divided into the training set, development set, and test set, and the model performance is evaluated on the test sets. The three datasets of legal text are evaluated, respectively. Statistics of the preprocessed dataset information are shown in Table 1.

5.3. Baselines. To verify the effectiveness of our proposed model, the results are compared with the following models.

(i) Binary Relevance (BR) [1] transforms the MLC task into multiple single-label classification problems by ignoring the correlations among labels

(ii) Classifier Chain (CC) [2] converts the MLC task into chains of binary classification in which first classifier is trained only on the input dataset, and then, each classifier is trained on all previous classifiers in the input space and chain by taking high-order label correlations into consideration

(iii) Label Powerset (LP) [18] transforms the multilabel classification problem into a multiclass problem where the classifier is trained on all unique label combinations found in the training dataset

(iv) CNN [12] applies multikernels to extract features of text, which are then feeded into fully connection layer with sigmoid function to capture the label probability distribution of label

(v) CNN-RNN [8] applies CNN and RNN to capture local semantic information and global semantic information and models the correlation among labels

(vi) SGM [9] proposes a sequence attention-based generation model with a new decoder structure to solve the problem of multilabel classification and models the correlation among labels

(vii) SU4MLC [17] generates text local representations with multidilated convolution and attention mechanisms to generate the maximum probability sequence of labels

(viii) ML-Reasoner [22] proposes a new iterative method to focus on label information and applies a binary classifier to predict all labels at the same time

5.4. Experiment Setup. We conduct our experiments with PyTorch on the Nvidia Titan-V GPU. The batch size is set to 16 on RCV1V2 dataset and 32 on other legal text datasets, and the size of both word embedding is set to 512 on all datasets with random initialization. The hidden sizes of LSTM in the encoder and decoder are 512, and the number of LSTM layers in the encoder and LSTM layer in the decoder is 2 and 2. The kernel sizes of convolution in encoder are [1, 3, 5] and [2, 3, 5] on RCV1V2 dataset and legal text datasets. ($\alpha = 0.95$, $\beta = 0.05$) and ($\alpha = 0.75$, $\beta = 0.25$) on RCV1V2 dataset and legal text datasets. The number of convolution filters is 512. To avoid overfitting, we employ the dropout [30]. The initial learning rate is 0.0003, and it is drop as the epoch changes.

5.5. Evaluation Metrics. Following the previous work [8, 9], we measure hamming loss [31] and micro-F1 score [32] as our main evaluation metrics. For reference, microprecision and microrecall are also reported.

(i) *HammingLoss*. HammingLoss evaluates the fraction of misclassified instance-label pairs, where a relevant label is missed or an irrelevant is predicted, which is calculated as follows:

$$\text{HammingLoss} = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{XOR}\left(Y_{i,j}, P_{i,j}\right)}{L}, \tag{21}$$

where N is the number of samples, L is the number of labels, $Y_{i,j}$ is the true value corresponding to the j-th label of the i-th prediction, $P_{i,j}$ is the predicted value corresponding to the j-th label of the i-th prediction, and $\text{XOR}(0, 1) = \text{XOR}(1, 0) = 1$.

(ii) *Micro-F1*. Micro-F1 can be interpreted as a weighted average of the precision and recall. It is calculated globally by counting the total true positives tp_j, false negatives fn_j, and false positives fp_j, which is calculated as follows:

$$\text{Micro} - F1 = \frac{\sum_{j=1}^{L} 2\text{tp}_j}{\sum_{j=1}^{L} 2\text{tp}_j + \text{fp}_j + \text{fn}_j}. \tag{22}$$

5.6. Experiment Results

(i) *RCV1-v2 Dataset*. The experimental results on RCV1-v2 dataset are shown in Table 2. Compared to with the baselines, our model ranks third in the hamming loss with 0.00817 and second in the micro-F1 with 87.55% among the methods in Table 2. Compared with the best model (LP) in baseline based on traditional machine learning, our approach achieves an improvement of 2.04% micro-

Table 1: Statistics of dataset.

Dataset	Total labels	Sample of train set	Sample of validation set	Sample of test set	Words/sample	Label/sample
RCV1V2	103	802414	1000	1000	123.94	3.24
Divorce	20	12727	3927	3932	104.28	1.75
Loan	20	6384	1993	1975	169.55	1.93
Labor	20	6045	1855	1837	131.28	1.51

Table 2: The results on RCV1-v2 dataset test dataset. HL represents hamming loss. The symbol "+" denotes that the higher the value is, the better the model performs. The symbol "-" is the opposite.

Models	HL (-)	Micro-P (+)	Micro-R (+)	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
CNN-RNN	0.0085	0.889	0.825	0.856
SGM	0.0082	0.897	0.835	0.864
SU4MLC	0.0077	0.8868	0.8631	0.8748
ML-Reasoner	0.0081	0.912	0.847	0.878
Our model	0.00817	0.8831	0.8682	0.8755

F1 score and a reduction of 5.0% hamming loss on RCV1-v2 test dataset. Compared with the best baseline model based on deep learning, our approach is competitive on RCV1-v2 test dataset.

(ii) *Divorce Dataset*. The experimental results of our model compared with the baselines on divorce dataset are shown in Table 3. Compared with the baselines, our model ranks first in the hamming loss with 0.02126 and micro-F1 with 87.90% among the methods in Table 3

Compared to the best baseline model, our approach achieves a reduction of 3.26% hamming loss and an improvement of 0.65% micro-F1 score on divorce test dataset.

(iii) *Loan Dataset*. The experimental results on loan dataset are shown in Table 4. Compared with the baselines, our model ranks first in micro-F1 with 86.03% and ranks second in the hamming loss with 0.02772 among the methods

Compared with the best baseline model, our approach achieves an improvement of 1.06% micro-F1 score on loan test dataset.

(iv) *Labor Dataset*. The experimental results on loan dataset are shown in Table 5. Compared with the baselines, our model ranks first in the hamming loss with 0.01689 and micro-F1 with 86.99% among the methods in Table 5

Compared with the best model in baseline model, our approach achieves a reduction of 1.56% hamming loss and an improvement of 0.05% micro-F1 score on labor test dataset.

Since our proposed model can obtain more important local semantic information using multilevel CNN network than single convolution and dilated convolution and applies the multiattention method to integrate local feature information obtained by multi-CNN, context information captured by LSTM, and label sequence information in the decoding process, our model outperforms the baseline models on both general text datasets and legal text datasets.

5.7. Ablation Test. To evaluate the effects of the modules in our proposed model, we perform ablation tests on our model. We analyze the results of CNN layer number and initialized fully connection layer with different initialization method on three test sets.

5.7.1. The Impact of CNN Layer Number. In order to verify the influence of the number of convolution layers in our model on the effect of the model, we conduct comparative tests on legal text datasets, respectively. During the coding period, we extract the local feature information around words by using multilevel convolution without maxpooling layer and single convolution layer without maxpooling layer, respectively, as the local feature representation of words. In the single convolutional network, we use convolution kernels of sizes 2, 3, and 5 to capture local semantic representation, respectively, while we use the convolutional kernel (kernel size = 2,3,5) in the multilevel convolution network to extract long distance local semantic representation of words. The experimental comparison results are shown in Table 6.

The results in Table 6 show that, in terms of legal text datasets, the results of our model are still better than those of the single CNN model, which indicates that multikernels with different sizes can capture abundant *n*-gram information with rich semantics and generate better local semantic features of words than single kernel for legal texts where the number of characters that make up a word is uncertain and the text length is relatively short.

5.7.2. The Impact of Fully Connection Layer with Frequency Initialization (FCFI). In the decoding process, we employ a fully connection layer with initialization to pay attention to the correlation between any two labels. To evaluate the effectiveness of our initialization approach, we construct the model without fully connection layer and model initialized using [25] separately. We apply the aforementioned two

Table 3: The results on *Divorce* test dataset.

Models	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
CNN	0.02516821	0.8624	0.8453	0.8537
CNN-RNN	0.02517237	0.8657	0.8532	0.8594
SGM	0.02197737	0.8810	0.8626	0.8717
SU4MLC	0.02497054	0.8796	0.8541	0.8667
ML-Reasoner	0.02631581	0.8901	0.8573	0.8733
Our model	0.02126052	0.8852	0.8728	0.8790

Table 4: The results on *Loan* dataset test dataset.

Models	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
CNN	0.03033127	0.8418	0.8397	0.8407
CNN-RNN	0.03015758	0.8369	0.8406	0.8387
SGM	0.02958177	0.8432	0.8370	0.8401
SU4MLC	0.02883519	0.8555	0.8227	0.8388
ML-Reasoner	0.01861665	0.8719	0.8317	0.8513
Our model	0.02772556	0.8633	0.8573	0.8603

Table 5: The results on *Labor* dataset test dataset.

Models	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
CNN	0.02499071	0.8469	0.8275	0.8371
CNN-RNN	0.02551891	0.8497	0.8365	0.8430
SGM	0.01820437	0.8740	0.8451	0.8593
SU4MLC	0.01758433	0.8796	0.8487	0.8639
ML-Reasoner	0.01715832	0.8759	0.8633	0.8695
Our model	0.01688988	0.8812	0.8589	0.8699

methods to conduct experiments on the legal text datasets. The experimental results of three models are shown in Table 7. The symbol IFC, w/o, no-order represents the model with initialization using [25], without the initialized fully connection layer and our model, respectively. From Table 7, it can be seen that compared with the model without initialized layer, the model with our initialized method on or reduces the hamming loss by 9.53%, 8.27%, and 43.88% and improves the micro-F1 by 1.71%, 2.70%, and 3.41%, on the divorce, loan, and labor datasets with label order, respectively, and compared with the model with initialization using [25], the model with our initialized method reduces the hamming loss by 4.1%, 2.43%, and 13.4% and improves the micro-F1 by 0.81%, 1.44%, and 0.90% on the divorce, loan, and labor datasets with label order, respectively. The experimental results of our proposed model also exceed that of IFC and w/o models, indicating that the introduction of normalization in the initialization can improve the performance of the model.

It can be seen from Table 7. The performances of the methods proposed by [25] and us all exceed the performance of the method without initialization of parameters of the full connection layer, indicating that initialization of parameters of the full connection layer is conducive to improving the classification effect of the model. According to our analysis, the reason why the result of our method exceeds that of the method proposed by [25] is that when the method in [25] initializes parameters, the calculation of $F(i, j)$ only considers the calculation of the inner part of the i-th row and ignores the effect of the j-th row. Our method considers the correlation between the i-th row and j-th row according to Formula (17).

5.7.3. The Impact of the Input Embedding of Each Levels in MCNN. When CNN is used to extract local neighbor information of text, multilayer convolution structure is used in this paper. In the MCNN proposed by us, the input of each layer takes the output result of the bottom CNN as the input of the top CNN, among which the CNN input of the bottom layer is the embedding vector of the text. In order to verify the effectiveness of our approach, we constructed a multilayer CNN network with multiple different convolution kernels. In this structure, the input of CNN at each layer is the embedding vector of text. We also conducted experiments on three legal datasets. The experimental results of two models are shown in Table 8. The symbol "TSI" represents the model, and the input of CNN at each layer is the same, which is the text embedding vector. From Table 8, it can be seen that compared with the TSI model, the model with our method reduces the hamming loss by 2.97%, 1.91%, and 1.80% and improves the micro-F1 by 1.54%, 1.14%, and 1.07% on the divorce, loan, and labor datasets.

The results show that the proposed method of using the output of lower-layer CNN as the input of upper-layer CNN can improve the effect of the model.

5.7.4. The Impact of the Concatenated Method of Outputs of Different CNN Layers. Since different convolution kernels and multilevel CNN can extract different local semantic features of the text more effectively, we concatenate the output results of CNN at all layers to form the final feature output of MCNN. In order to verify the effectiveness of our proposed method, we constructed a model with the top-level CNN output as the final output of MCNN model and carried out experiments on three legal datasets, respectively. The experimental results of two models are shown in Table 9. The w/o represents the model with the top-level CNN output as the final output of MCNN model. From Table 9, it can be seen that compared with the w/o model, the model with our method reduces the hamming loss by 5.8%, 4.7%, and 10% and improves the micro-F1 by 1.82%, 2.03%, and 2.98% on the divorce, loan, and labor datasets.

Experimental results show that the proposed concatenation of outputs of different CNN layers as the final output of MCNN results can improve the effect of the model.

Table 6: The comparison results on legal dataset of single-layer CNN and multilevel CNN.

Model	Divorce				Loan				Labor			
	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
Single CNN ($k = 2$)	0.02337	0.8776	0.8612	0.8693	0.02814	0.8468	0.8416	0.8441	0.01785	0.8757	0.8386	0.8567
Single CNN ($k = 3$)	0.02280	0.8703	0.8705	0.8704	0.02892	0.8482	0.8386	0.8434	0.01904	0.8617	0.8461	0.8538
Single CNN ($k = 5$)	0.02385	0.8652	0.8719	0.8685	0.02762	0.8512	0.8296	0.8402	0.02131	0.8576	0.8432	0.8503
Our model	0.02126	0.8852	0.8728	0.8790	0.02772	0.8633	0.8573	0.8603	0.01688	0.8812	0.8589	0.8699

Table 7: The comparison results on legal dataset of fully connection layer.

Model	Divorce				Loan				Labor			
	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
w/o	0.02350	0.8649	0.8635	0.8642	0.03022	0.8358	0.8396	0.8377	0.03008	0.8254	0.8575	0.8412
IFC	0.02217	0.8702	0.8735	0.8719	0.02841	0.8424	0.8537	0.8481	0.01949	0.8696	0.8548	0.8621
No-order	0.02126	0.8852	0.8728	0.8790	0.02772	0.8633	0.8573	0.8603	0.01688	0.8812	0.8589	0.8699

Table 8: The comparison results on legal dataset of the input embedding of each levels in MCNN.

Model	Divorce				Loan				Labor			
	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
TSI	0.02191	0.8799	0.8519	0.8657	0.02826	0.8569	0.8443	0.8506	0.01719	0.8710	0.8507	0.8607
Our model	0.02126	0.8852	0.8728	0.8790	0.02772	0.8633	0.8573	0.8603	0.01688	0.8812	0.8589	0.8699

Table 9: The comparison results on legal dataset of the concatenated method of outputs of different CNN layers.

Model	Divorce				Loan				Labor			
	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)	HL (-)	Micro-P (+)	Micro-R (+)	Micro-F1 (+)
w/o	0.02257	0.8676	0.8591	0.8633	0.02910	0.8479	0.8385	0.8432	0.01876	0.8599	0.8301	0.8447
Our model	0.02126	0.8852	0.8728	0.8790	0.02772	0.8633	0.8573	0.8603	0.01688	0.8812	0.8589	0.8699

6. Conclusion

In this paper, we propose a model based on multilevel convolutional network for sentence-level element extraction of legal text. Our proposed model can combine textual local semantic information obtained by the multilevel CNN and context information obtained by LSTM to generate higher level semantic representation of sentences by applying multiattention network. The initialization method we proposed can improve the classification effect of the model by normalizing the cooccurrence relationship between labels. Experimental results on a general text dataset and three legal domain datasets demonstrate that our model achieves the expected results in the evaluation metrics by comparing with the baseline model. By comparing the results with single-layer CNN on the legal text dataset, our proposed multilevel CNN is more capable of extracting the semantic features of the legal text by applying different window sizes of convolution kernels to alleviate the number of characters in each word in the legal text is not fixed problems. In addition, by comparing the initialization method with other initialization methods, our proposed initialization method can make a contribution to improving multilabel classification task of the legal text.

Data Availability

The processed legal text data used to support the findings of this study are currently under embargo, while the research findings are being commercialized. Requests for data 6–12 months after the publication of this article will be considered by the corresponding author.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2018YFC0830900).

References

[1] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[2] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.

[3] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2001.

[4] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Advances in neural information processing systems*, vol. 14, pp. 681–687, 2002.

[5] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[6] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[7] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5413–5423, 2017.

[8] 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 *2017 International joint conference on neural networks (IJCNN)*, pp. 2377–2383, IEEE, 2017.

[9] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: sequence generation model for multi-label classification," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*pp. 3915–3926, Santa Fe, New Mexico, USA, 2018.

[10] Z. Li, Z. Yang, Y. Xiang, L. Luo, Y. Sun, and H. Lin, "Exploiting sequence labeling framework to extract document-level relations from biomedical texts," *BMC Bioinformatics*, vol. 21, no. 1, p. 125, 2020.

[11] Z. Yang and G. Liu, "Hierarchical sequence-to-sequence model for multi-label text classification," *IEEE Access*, vol. 7, pp. 153012–153020, 2019.

[12] G. Kurata, B. Xiang, and B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 521–526, Association for Computational Linguistics, San Diego, California, 2016.

[13] Y. Liang, H. Li, B. Guo et al., "Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification," *Information Sciences*, vol. 548, pp. 295–312, 2021.

[14] S. M. A. Shah, H. Ge, S. A. Haider et al., "A quantum spatial graph convolutional network for text classification," *Computer Systems Science and Engineering*, vol. 36, no. 2, pp. 369–382, 2021.

[15] H. Wang, K. Tian, Z. Wu, and L. Wang, "A short text classification method based on convolutional neural network and semantic extension," *International Journal of Computational Intelligence Systems*, vol. 14, pp. 367–375, 2021.

[16] W. Yang, J. Li, F. Fukumoto, and Y. Ye, "HSCNN: a hybrid-siamese convolutional neural network for extremely imbalanced multi-label text classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pp. 6716–6722, Association for Computational Linguistics, 2020.

[17] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, "Semantic-unit-based dilated convolution for multi-label text classification," in *Conference on Empirical Methods in Natural Language Processing*, 2018.

[18] G. Tsoumakas and I. Katakis, "Multi-label classification: an overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2009.

[19] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[20] C. Li, B. Wang, V. Pavlu, and J. Aslam, "Conditional bernoulli mixtures for multi-label classification," in *International conference on machine learning*, pp. 2482–2491, 2016.

[21] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification—revisiting neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 437–452, Springer, 2014.

[22] R. Wang, R. Ridley, X. Su, W. Qu, and X. Dai, "A novel reasoning mechanism for multi-label text classification," *Information Processing and Management*, vol. 58, no. 2, p. 102441, 2021.

[23] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pp. 655–665, 2014.

[24] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[25] W. Liao, Y. Wang, Y. Yin, X. Zhang, and P. Ma, "Improved sequence generation model for multi-label classification via CNN and initialized fully connection," *Neurocomputing*, vol. 382, pp. 188–195, 2020.

[26] P. Wang, B. Xu, J. Xu, G. Tian, C. Liu, and H. Hao, "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification," *Neurocomputing*, vol. 174, pp. 806–814, 2016.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] J. Deng, L. Cheng, and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification," *Computer Speech & Language.*, vol. 68, article 101182, 2021.

[29] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: a new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[32] D. Christopher, "Manning, introduction to information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 43, no. 3, pp. 824-825, 2008.