IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 15, 2019, accepted July 23, 2019, date of publication August 5, 2019, date of current version August 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933042

MsCoa: Multi-Step Co-Attention Model for Multi-Label Classification

HAOYANG MA, YUJUN LI, XIANPENG JI, JUNLEI HAN, AND ZEQIANG LI
School of Information Science and Engineering, Shandong University, Qingdao 266200, China

Corresponding author: Yujun Li (liyujun@sdu.edu.cn)

This work was supported by the Key Research and Development Program of China under Grant 2018YFC0831000.

ABSTRACT Multi-label text classification (MLC) task, as one of the sub-tasks of natural language processing, has broad application prospects. On the basis of studying the previous research work, this research takes the relationship among text information, leading label information and predictive label information as the frame and analyzes the information loss of original text and leading label, decoding error accumulation. We propose an improved multi-step multi-classification model to mitigate the phenomenon of error prediction, label repetition and error accumulation. The model uses multi-step and multi-classification task to complete multi-label prediction. It uses the leading label and the original text as input, and the next to-be-predicted label as output. The co-attention mechanism is operated between the original text and the leading label. The attention of the original text to the leading label is helpful to filter out the error accumulation problem caused by the error prediction. The features is combined in a manner of difference and concatenation, which highlights the auxiliary effect of the model structure on feature extraction. In order to avoid the influence of the feature dimension on the performance of long short-term memory (LSTM), a multi-layer fully-connected classifier is used instead to predict the label. Through experimental validation, the performance of our model on the multi-label text classification task shows the current optimal level, which fully proves the superiority of our model.

INDEX TERMS Artificial neural network, attention mechanism, deep learning, multi-label classification.

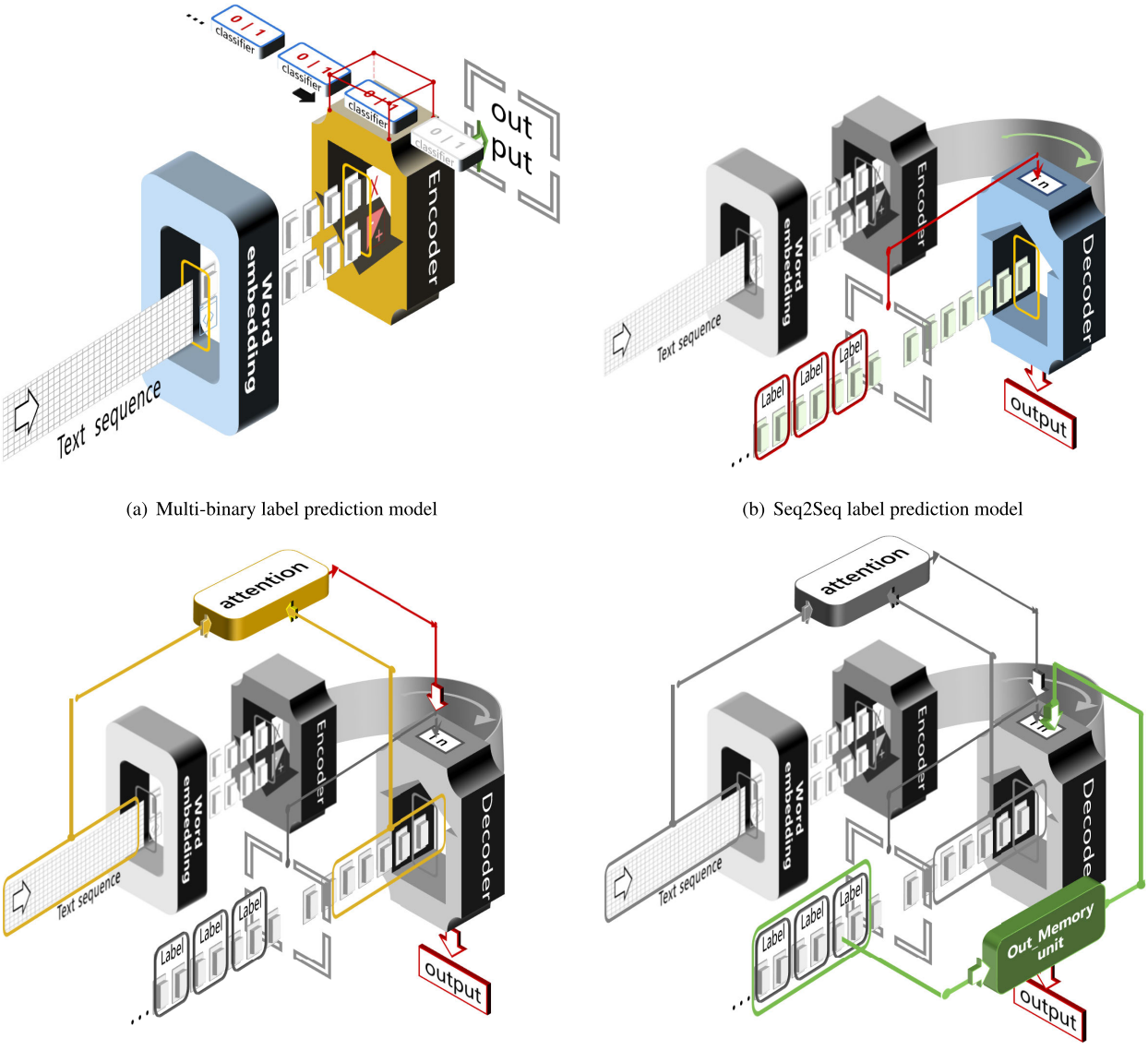
I. INTRODUCTION

The multi-label text classification, as a sub-field of natural language processing, has broad application prospects in the fields of recommendation system [1], information retrieval [2], text categorization [3], [4] and so on. Unlike multiple instances sharing one label [5], multi-label learning allows one instance to correspond to multiple labels. At present, the text classification method based on deep neural network can show the best performance. Based on the progress of the basic technology, the multi-label text classification task can also be solved by deep artificial neural network. In this stage of development, a wide variety of multi-label text classification methods have emerged, which can be attributed to the following two categories.

The first method is called transformation method, including task transformation and label transformation. The basic idea of the transformation method is to transform the multi-label text classification task into a single-label classification task. The representative method of the task transformation model is the multi-binary classification model proposed by Matthew R. Boutell [6] et al. in 2014, which constructs a binary classification model on each category of labels. Then the multi-label classification task is transformed into a multiple binary classification prediction task, which is similar to the multi-task task. The basic structure of this model is shown in Figs. 1-a. This kind of model sets up the semantic association between the text and the label, but it neglects the association between the labels and can not further improve the performance. The idea of label transformation method is to construct the label sequence as a new single label, but this transformation method is not suitable for cases of large number of label classes, because original labels have exponential number of combinations.

The second method is called algorithm adaptation, including the k nearest neighbor algorithm [7] and the sequence generation model based on the sequence to sequence (Seq2Seq) framework. The multi-label classification model under the Seq2Seq framework realizes the multi-label prediction by using the label sequence generation method, and pays attention to the semantic association between the labels [8]. The basic structure of this type of model

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



(a) Multi-binary label prediction model

(b) Seq2Seq label prediction model

(c) Seq2Seq+attention label prediction model

(d) Seq2Seq+attention+out-memory-unit label prediction model

FIGURE 1. Typical multi-label text classification models. Note that the newly added modules are colored.

is shown in Figs. 1-b.After that, many researchers began a number of improvements based on this framework, and Lin *et al.* improved the encoder structure of the Seq2Seq framework [9], increasing the embedding effect of the encoder on the phrases, and Yang *et al.* introduced the global embedding unit in the decoder [10]. However, there is a problem of gradient disappearance in the Seq2Seq model with RNN [11], leading to information loss of original text and leading label. Some research work draws on the attention mechanism [12]–[14] in machine translation to improve the performance of the model, and obtains information supplements from the original text [15]–[17]. To some extent, the problem of the loss of original text information in encoder and decoder is alleviated. The model is shown in Figs. 1-c. Other studies introduced an out-memory-unit [18] in the decoder based on the attention mechanism. These methods emphasize the already predicted label to the decoder [19], and alleviate the problem of leading label information loss caused by the disappearance of the gradient. The model with an out-memory-unit is shown in Figs. 1-d.

Based on the above work, we propose an improved model of multi-step multi-classification prediction. The main contributions of this study can be summarized as follows:

- *In this paper, we divide the label sequence into the leading label and the next to-be-predicted label by preprocessing the original data. It can meet the training needs of multi-step prediction model. By introducing the co-attention mechanism between the original text and the leading label, the information filtering function of the leading label in the process of text coding is realized, and the training process is optimized. The attention effect of the original text content on the leading*

label further alleviates the error accumulation problem caused by single error prediction.

- *According to the characteristics of multi-label text classification task, our model adopts a manner of difference and concatenation for feature vector. The method of difference highlights the original text information on which the to-be-predicted label depends and optimizes the supervision effect of the label information, and finally realizes the modeling of the original text, the leading label and to-be-predicted label.*

This paper is organized as follows: In section II, we generalize the current researchers' work, and further deduce the theoretical direction of the performance improvement effect. Then, our model structure based on this theory and method are introduced. In section III, the experimental validation and model performance evaluation are presented. Section IV summarizes the research work and analyzes the remaining problems in the present work. Finally, we look forward to the future research direction and technology development prospect in the field of multi-label text classification.

II. RESEARCH ON MULTI-LABEL TEXT CLASSIFICATION ALGORITHMS

There are many text classification algorithms, such as multi-label decision tree (ML-DT) [20], Rank-SVM [21], label powerset (LP), multi-label k nearest neighbor (ML-KNN) [22], [23], classifier chains (CC) [24], improved version based on CC [25], [26], and so on.

For the task of multi-label classification, which aims at mining text information from many aspects and providing multi-angle information service, there is a parallel association between the labels in the label collection, which is different from the branch-hierarchy relationship. It is also different from the sequential relationship between words similar to that in natural languages. Therefore, there is no serialization structure between the labels of multi-label text classification on semantic information. For example, the labels with ''politics, economy, culture'' and ''economy, politics, culture'' have the same text content as those defined by the ''economic, political, cultural'' labels.

However, after the label sequence in the original data is processed by manual rules, the mapping relationship between the text information and the labels to be predicted is not fundamentally different, so that the set of labels to be predicted is transformed into a sequence of labels to be predicted. Therefore, serialization processing can be applied to multi-label text classification tasks. The original text is serialized data, and labels become serialized data after processing. It is helpful to use sequence generation to consider the semantic association among labels. As discussed in the introduction, the performance of the sequence generation method fully proves the feasibility of the serialization processing of the labels. Therefore, it makes sense to use serialization to preprocess label sequences [27].

Based on this rationality, the current multi-label classification model is generally encoder-decoder structure, using Seq2Seq framework to implement label sequence generation to be predicted. In this process, the serialization modeling method with LSTM as the basic function unit is widely used in this field. In part of the research work, the serialization modeling of encoder adopts convolution neural network method to grasp more semantic unit information [28]–[31]. And the label generation process of decoder is usually based on recurrent neural network. Under the assumption of idealized model, the serialization of labels transforms complex multi-label classification tasks into sequence generation tasks and the feasibility of multi-label classification is greatly simplified.

A. THEORETICAL FRAMEWORK OF MULTI-LABEL TEXT CLASSIFICATION

The main elements of the multi-label text prediction are the information of original text, the leading label and the label to be predicted. Based on these three elements, the multi-label text classification task can be decomposed into original text information extraction, leading label information extraction and the new label prediction three basic tasks.

Under the above theoretical framework, the three problems of error prediction, repeated prediction and error accumulation mentioned in this paper can be respectively attributed to the loss of original text information, leading label information loss and error accumulation. The information loss of the original text and the leading label is directly caused by the gradient disappearance problem of the RNN, and the error accumulation is caused by the sequence decoding mode of the RNN. Therefore, the key to improve the model effect is to solve the above problems. In essence, the attention mechanism of encoder-decoder improves the performance by solving the problem of original text information loss, and the method of out-memory-unit improves the performance by solving the problem of leading label information loss. The problem of error accumulation is the remaining problem of current research work.

Based on the previous research work, we provide a improved model to solve the problem of the error accumulation of the leading label information, and introduce a training method to adapt to the structure of the model. The nature of the multi-label text classification problem is to predict new label based on the information of the context text and the leading label, therefore, the method is applicable to the end-to-end model structure. At the same time, the prediction of multiple labels can be realized by multiple feedforward calculations and updating the leading labels.

B. THE THEORETICAL DERIVATION OF THE FORMALIZATION

In this section, we conclude the mathematical model of previous research work through formula derivation. Combined with the information loss of original text and the leading label and the error accumulation, we discuss the progress

and shortcomings of these work and propose improvement strategies. Finally, based on solving the above three problems, we proposes an improved formulation and introduces our model.

The multi-label text classification task can be generalized to the mapping of the text sequence to the label sequence, and the probability learning model can be defined as:

$$P = P(S_L \mid S_T), \tag{1}$$

where S_T represents a text sequence$\{x_1, x_2, \ldots, x_m\}$ and S_L represents a label sequence$\{l_1, l_2, \ldots, l_m\}$. P is the probability of S_L under S_T.

The label transformation method attempts to implement the probability formula directly. The various combinations in the label set $\{L_0, L_1, L_2, \cdots, L_N\}$ make up new labels. However, a large number of new labels will cause data sparsity problems, thus reducing the applicability of the model.

The multi-binary classification model transformation strategy transforms the problem into modeling $P_i = P(L_i \mid S_T)$, where P_i represents the probability of the i_{th} label L_i under S_T. This strategy calculated the probability of each label, and did not take into account the semantic correlation between the labels, which limited the further improvement of the model effect.

Different from the multi-binary classification model, the sequential generation model takes into account the semantic correlation between the labels. The model formula is as follows:

$$P_i^{t+1} = P(L_i \mid Y^t, S_T), \tag{2}$$

where P_i^{t+1} denotes the probability of label L_i at time-step $t+1$, Y^t is the leading label at time-step t. This process involves two operations: sequence encoding and sequence generation.

In the implementation of the Seq2Seq framework, the idealized formula can be expressed as the following formula:

$$P_i^{t+1} = P(L_i \mid D^t(E(S_T), Y^t)), \tag{3}$$

where $E(S_T)$ represents the output of encoder in Seq2Seq model which takes S_T as input. And D^t denotes the output of decoder which takes $E(S_T)$ and Y^t as input. The decoder at time-step t predicts the next label based on the original text and the leading label. In this process, the original text information will be lost in the forward propagation, and will be reduced to insufficient text information $\tilde{E}(S_T)$. Similarly, in the process of sequence decoding, the decoded sequence is re-input to the decoder, and there is also a phenomenon of information loss, and the leading label information is reduced to $\tilde{Y}^t$. That is, the multi-label text prediction model under the Seq2Seq framework degenerates from the idealized form to the following formula:

$$P_i^{t+1} \approx P(L_i \mid D^t(\tilde{E}(S_T), \overset{\sim}{Y^t})). \tag{4}$$

Similar to the machine translation model, after introducing the attention mechanism across the encoder and decoder, the formula is as follows:

$$P_i^{t+1} \approx P(L_i \mid D^t(\tilde{E}(S_T), A_{hT}^t, \tilde{Y}^t)), \tag{5}$$

where A_{hT}^t denotes the attention vector of the original text on the label at time-step t. By introducing such an attention mechanism, the information with attention of original text is re-inputted to the decoder, and the loss of information in the forward propagation process is alleviated. However, the problem of information loss of the leading label sequence has not been solved in the decoding process.

An out-memory-unit is introduced into the decoder to encode the leading label in the decoding process. And its probability formula is as follows:

$$P_i^{t+1} \approx P(L_i \mid D^t(\tilde{E}(S_T), A_{hT}^t, \tilde{Y}^t, M_Y^t)), \tag{6}$$

where M_Y^t represents the out-memory-unit of all the predicted leading labels before time-step t, and it enhances the information of the leading label. However, there is a certain contradiction between serialization prediction process and out-memory-unit. Once a error prediction occurs in the process of prediction, the result of this error prediction will be memorized. It is magnified gradually in the subsequent sequence decoding process, and then affects the accuracy of the model in the prediction. A error memory will cause the above formula to be reduced to:

$$P_i^{t+1} \approx P(L_i \mid D^t(\tilde{E}(S_T), A_{hT}^t, \tilde{Y}^t, \tilde{M}_Y^t)), \tag{7}$$

where $\tilde{M}_Y^t$ is degraded from M_Y^t after inputting erroneous prediction information in the out-memory-unit.

Based on the above discussion, it is the key to solve the problem of error accumulation in decoding process. We let the original text information conduct information filtering on the leading label, and eliminate the error information that contradicts the original text information. Then we provide the filtered leading label information to the decoder. The decoder can be less affected by a previous error prediction, and then improve the accuracy of prediction in the decoding process. The probability formula for this method is as follows:

$$\begin{aligned} P_i^{t+1} &= P(L_i \mid A_{YT}^t, A_{TY}^t, H), \\ P &= \prod_{t=1}^{M+1} \max(P_i^t), \end{aligned} \tag{8}$$

where A_{YT}^t and A_{TY}^t represent the attention information of the leading label to the original text and the original text to the leading label at time-step t, respectively. H represents the hidden state output of the encoder which contains the original text information. P represents the joint probability of all the labels. In the prediction process, A_{YT}^t needs to be removed because it represents the text information corresponding to the predicted leading label.

Based on the following three reasons, a multi-step discriminant end-to-end model is used to replace the sequence generation model in the label prediction phase:

Multi-label

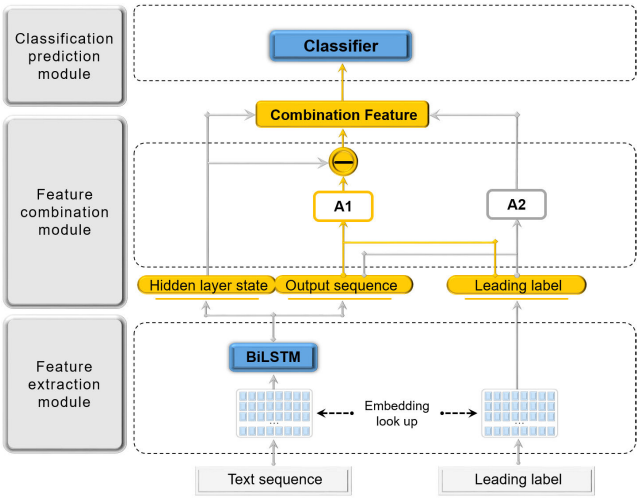


FIGURE 2. The structure diagram of the model.

1 text classification is different from pure sequence prediction. Although the label prediction needs to be predicted orderly, the leading label does not have the order problem. Once a label is predicted and used as a leading label, there is no need for sequencing. Therefore, in essence, the leading label does not need to be input sequentially into the decoder for forward propagation, and the generated model will complicate the problem.

2 The support of the original text information to each prediction label is related to the text meaning and the label meaning, but independent of the label order. In the prediction of each label, the same original text information should be adopted. Therefore, the original text information also does not need to be decoded through the sequence of forward propagation process.

3 Multi-dimensional feature vectors constructed by concatenation will have very large dimensions. If this decoding vector is input into the LSTM cell, there is a contradiction between the LSTM performance and its input scale, that is, when the input scale of LSTM increases to a certain extent, the serialization modeling ability of LSTM has a negative correlation with its input scale. Therefore, in order to coordinate the contradiction between high-dimensional features and RNN series decoders, the decoder is not suitable for using the neural network structure of sequence generation.

C. MODEL STRUCTURE

In this section, we introduce the structure of our model and the relationship among the modules. In addition, we also introduce our data preprocessing methods.

The overall structure of our model is shown in Fig. 2, where BiLSTM denotes bi-directional LSTM, $A1$ and $A2$ represent the attention vector of the leading label to the original text and the original text to the leading label, respectively. On the whole, the model can be divided into feature extraction module, feature combination module and classification prediction module. The feature extraction module extracts the original text feature and the leading label feature from the input, and the original text information includes the hidden state of the bi-directional LSTM and the output sequence. The feature combination module of the model sets the combination mode of the above three features and gives the feature vectors needed to predict the new labels. The classification prediction module realizes the prediction of the next label to be predicted through the classification function based on the combination feature vectors above.

Our model achieves multi-label classification through multiple single-label classification tasks. The model reuses parameters in multiple predictions and updates the leading labels after the end of one prediction, and then carries out a new round of prediction.

In a prediction process, our model takes the original text sequence and the leading label as the input, and takes the probability distribution of the prediction label as the output. First, the model embeds the original text and the leading labels with a startup label as the first leading label and an end label as the last label. Then the text vector sequence $\{x_1, x_2, \ldots, x_N\}$ and the leading label vector sequence $\{l_1, l_2, \ldots, l_M\}$ are obtained. After that, the text sequence $\{x_1, x_2, \ldots, x_N\}$ is inputted into a bi-directional LSTM [32] for encoding. To some extent, the output of the current time-step in bi-directional LSTM contains information that will appear after it [33], thus optimizing the information extraction capability of RNN. The hidden state h_N of the bi-directional LSTM and the output sequence $\{w_1, w_2, \ldots, w_N\}$ are used as the representation of the original text information. Then the co-attention operation is carried out between the output sequence $\{w_1, w_2, \ldots, w_N\}$ and the leading label sequence $\{l_1, l_2, \ldots, l_M\}$. $A1$ and $A2$ are obtained after co-attention operation. $A1$ represents the text information that the leading label depends on which is no longer needed to predict the new label. We remove it from the original text information, that is, $h_N - A1$, and obtain the new feature vector to predict the new label. $A2$ represents the leading label information under the condition of the text information. On the one hand, $A2$ represents the predicted labels which is a precondition to predict new labels, on the other hand, the error information in the leading label is filtered out through this attention mechanism, and the problem of error accumulation solved. Then we concatenate $h_N, h_N - A1, A2$ to obtain a representation vector of the input information, which contains all the information that predicts the next new label. Finally, a six-layer fully-connected neural network classifier was used to reconstruct the multi-dimensional representation information, and the score distribution of each label was given. Softmax operation was further used to convert the score into probability distribution, so as to realize the prediction of a new label.

In the training process, the model takes the cross entropy loss as the loss function and the one-hot form of the label as the monitoring information. After completing a prediction, the newly predicted label will be added to the new leading

label. Repeat the above operation and gradually realize the prediction with multi-step and multi-classification task until the last label is predicted.

1) FEATURE EXTRACTION MODULE

The input of the feature extraction module is the embedding representation of the original text $\{x_1, x_2, \ldots, x_N\}$ and the leading label $\{l_1, l_2, \ldots, l_M\}$. Due to the serializability of the text sequence, we use bi-directional LSTM to further the encoding operation, and give the state vector of the encoder hidden layer h_N and output sequence $\{w_1, w_2, \ldots, w_N\}$. The formula for bi-directional LSTM is as follows:

$$\begin{aligned} i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \\ f_t &= \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f), \\ o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o), \\ g_t &= \tanh(W_{gh}h_{t-1} + W_{gx}x_t + b_g), \\ c_t &= i_t \odot g_t + f_t \odot c_{t-1}, \\ h_t &= o_t \odot \tanh(c_t), \\ h_{bi} &= \left[\overrightarrow{h}, \overleftarrow{h}\right], \end{aligned} \tag{9}$$

where i_t, f_t and o_t represent input gate, forget gate, and output gate respectively. σ, tanh denote the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ and hyperbolic tangent function $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ respectively, and $\odot$ denotes element-wise multiplication. W,b represent weight matrices and bias vectors respectively. h_{t-1}, x_t represent hidden state at time-step $t-1$ and input at time-step t respectively. h_{bi} denotes concatenated vector representation of forward hidden state $\overrightarrow{h}$ and backward hidden state $\overleftarrow{h}$. The input $\{x_1, x_2, \ldots, x_N\}$ in positive order and reverse order are cascaded through the LSTM module to obtain the hidden state, and h_N and $\{w_1, w_2, \ldots, w_N\}$ are obtained. Leading labels do not have sequentiality, so we directly take the embedded vector set of leading labels as the feature vector of leading labels.

2) FEATURE COMBINATION MODULE

Feature combination module includes co-attention operation, difference operation and concatenation operation. Taking hidden state h_N and output sequence $\{w_1, w_2, \ldots, w_N\}$ as the input, the two feature vectors A_{YS} and A_{SY} with attention weight information are obtained by the co-attention operation of the output sequence and the leading label sequence $\{l_1, l_2, \ldots, l_M\}$. A_{YS} denotes the information corresponding to the leading label in the text, and A_{SY} denotes the correlation information between the leading label and to-be-predicted label. The formula of A_{YS} is as follows:

$$\begin{aligned} e_{i,j} &= v_a^T tanh(W_a l_i + U_a w_j), \\ \alpha_{i,j} &= \frac{\exp(e_{i,j})}{\sum_{j=1}^{N} \exp(e_{i,j})}, \\ A_{YS} &= \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_{i,j} w_j. \end{aligned} \tag{10}$$

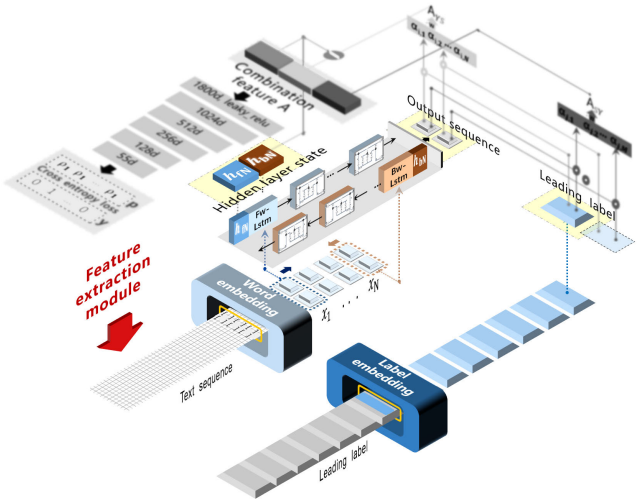


FIGURE 3. The structure of feature extraction module. Fw_LSTM denotes the forward LSTM, Bw_LSTM denotes the backward LSTM. h_{fN}, h_{bN} represent forward hidden state and backward hidden state at time-step N respectively. Note that the other modules are fuzzed.

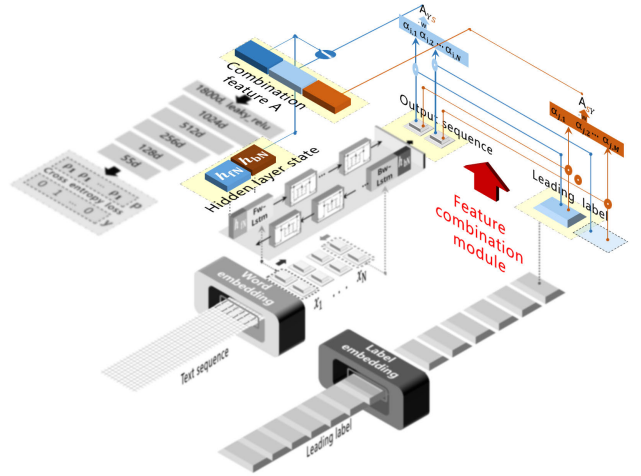


FIGURE 4. The structure of feature combination module.

The formula of A_{SY} is as follows:

$$\begin{aligned} e_{i,j} &= v_b^T tanh(W_b w_i + U_b l_j), \\ \alpha_{i,j} &= \frac{\exp(e_{i,j})}{\sum_{j=1}^{M} \exp(e_{i,j})}, \\ A_{SY} &= \frac{1}{M} \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{i,j} l_j, \end{aligned} \tag{11}$$

where W and U represent attention weight matrices. v^T is the context vector to distinguish informative elements from non-informative ones.

Finally, h_N, $h_N - A_{YS}$ and A_{SY} are cascaded to get the feature vector $A = \{h_N, h_N - A_{YS}, A_{SY}\}$.

3) CLASSIFICATION PREDICTION MODULE

The classification prediction module includes a multi-layer fully-connected neural network and a softmax layer.

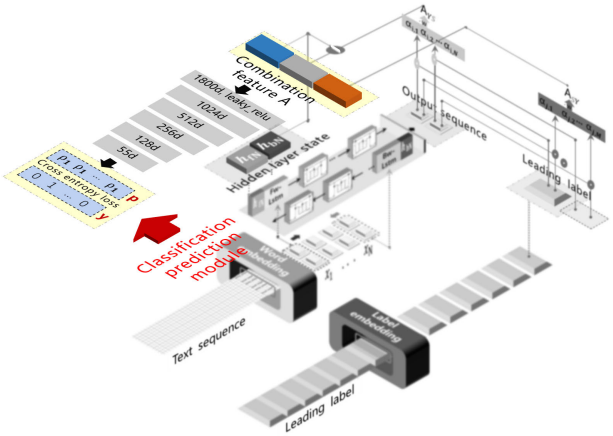


FIGURE 5. The structure of classification prediction module. Leaky_relu represents activation function.

The dimension of the first layer network is set to 1800, and then features are extracted in a descending manner. Finally, the probability distribution p of the predicted results in the label space is given. The formula for the probability p_i of the i_{th} label is as follows:

$$x = f(W v_A),$$
$$p_i = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}}, \tag{12}$$

where W, v_A represent all the weight matrices in fully connected layers in the module and the feature A obtained from feature combination module, respectively. f is the leaky relu activation function that has gradient in $\{\infty, 0\}$. K and x_i denote the number of all label classes and the value of the i_{th} label, respectively.

During the training, the loss between the probability distribution and the actual label should be calculated to optimize the model. We use cross entropy loss function in our model. The cross entropy function is given as follows:

$$j(\theta) = -\frac{1}{K}\sum_{i=1}^{K} y_i \log p_i + (1 - y_i)\log(1 - p_i), \tag{13}$$

where $y_i \in \{0, 1\}$ and $p_i \in \{0, 1\}$ are the true label value and predicted probability, respectively, for the i_{th} label. And K is the number of label classes.

The structure of classification prediction module is shown in Fig. 5.

It should be emphasized that the form of the original dataset is a text sequence as a sample, corresponding to a label sequence as a label. Different from Seq2Seq model, this model uses multi-step-prediction method to realize multi-label prediction, so we need to pre-process the original data. The label sequence adds the startup label and the end label respectively at the beginning and the end. Specifically, we combined the text with the startup label to predict the first real label. This predicted label is added to the leading label

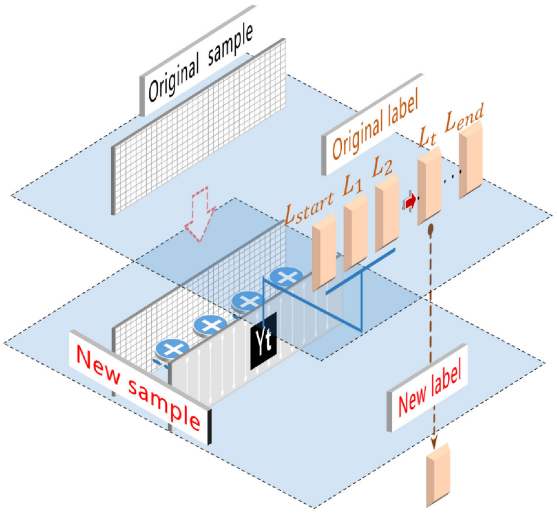


FIGURE 6. The mode of data parsing.

TABLE 1. Comparison between our model and all the baseline models on the AAPD test set.

Model	P	R	F1
BR	0.644	0.648	0.646
CC	0.657	0.651	0.654
LP	0.662	0.608	0.634
Seq2Seq+Attention	0.746	0.659	0.699
SGM+GE	0.748	0.675	0.710
Our Model(MSCoA)	**0.802**	**0.682**	**0.737**

sequence as a new member. The text is combined with the leading label to predict the next label until the end label is predicted. Therefore, an original sample will be parsed into $J + 1$ new samples, where J denotes the number of label sequence. We parse the original data in the form shown in the Fig. 6.

By doing this, we construct an agent task which converts a single-step multi-label classification to a multi-step single-label classification. Similarly, we can optimize our model by the loss of single-label classification task.

III. EXPERIMENT

In this section, we introduce our experimental work on two datasets to verify the performance and evaluation of our model. Firstly, we introduce the basic situation of datasets. And we introduce the contents of parsing the original dataset and constructing our training and test data. Then we introduce some parameters and operation settings in the process of training our model, and detail our training process. At the same time, the graphical representation of the loss function in the training process is given. Finally, we give accuracy and recall rate of multi-label text classification on the test dataset, and give the comparison between our model and some previous research work in Table 1 and Table 2. Taking the classification results of actual data as an example, the performance superiority of our model is demonstrated.

TABLE 2. Comparison between our model and all the baseline models on the RCV1-V2 test set.

Model	P	R	F1
BR	0.904	0.816	0.858
CC	0.887	0.828	0.857
LP	0.896	0.824	0.858
Seq2Seq+Attention	0.887	0.850	0.869
SGM+GE	0.897	0.860	0.878
Our Model(MSCoA)	**0.901**	**0.883**	**0.891**

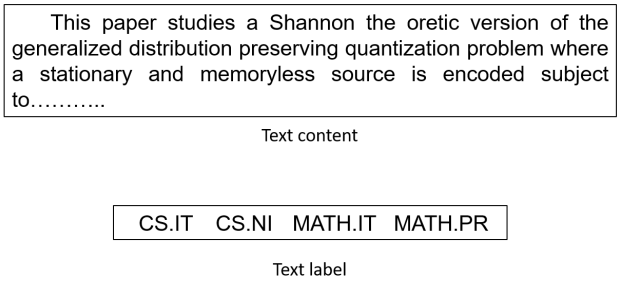


FIGURE 7. A typical dataset sample.

A. DATASETS

1) ARXIV ACADEMIC PAPER DATASET (AAPD)

This dataset is provided by Yang et al. and includes 55, 840 abstracts and corresponding subjects of papers in the computer science, with 54 kinds of label classes and contains 68, 767 words.

2) REUTERS CORPUS VOLUME I (RCV1-V2)

This dataset is provided by Lewis et al. It contains of over 800, 000 manually classified news samples made available by Reuters Ltd for research purposes. And there are 103 categories of labels in this dataset.

An example of a typical sample in a dataset is shown in Fig. 7.

Take AAPD dataset for example, we discard the samples whose text length was too long or too short, and randomly selected 40, 000 samples in the dataset. We build about 180, 000 new samples of data for model using the our dataset parsing method. 20% of these samples is used as the validation set to evaluate and the 80% is used as the training set. A portion of the data remains in its original form and is used to evaluate the performance of the classification of models after training.

B. MODEL CONSTRUCTION

We use pytorch to build our model and implement model data interface based on python. The model is structured as detailed in the previous section. Including word vectors and label vectors, all parameters of the model are initialized randomly by a standard initialization method with a mean value of 0 and a variance of 0.1, in which the word vector list is a matrix of $N * 300$. N denotes the size of our word dictionary. The label vector scale is a matrix of $55 * 600$, in which 54 items are normal label embedding vectors and one item is the startup label embedding vector. As an input leading label, the end label is not required.

In order to measure the model performance, we use the cross entropy between the output probability distribution and the real label in one-hot form as the agent loss function.

C. MODEL TRAINING AND EVALUATION

We build and train the model on a 64-gigabyte workstation with four TitanXP GPU. We use adam optimizer [34] to calculate gradient and update parameter in back propagation and the initial learning rate is set to 0.01. In training process, we randomly take 512 samples from the training set as a batch input to the model, and parallel computation on four GPU to carry out a forward propagation and error back propagation operation. After every 100 rounds of training, we evaluate the loss and accuracy on the validation set, and then continue the above iterative training.

In the training process, we record the model loss and prediction accuracy in the form of our analytic dataset to monitor the training process. The accuracy and the loss of validation set are shown in Figs. 8. And the hamming loss in the test process on AAPD is shown in Fig. 9.

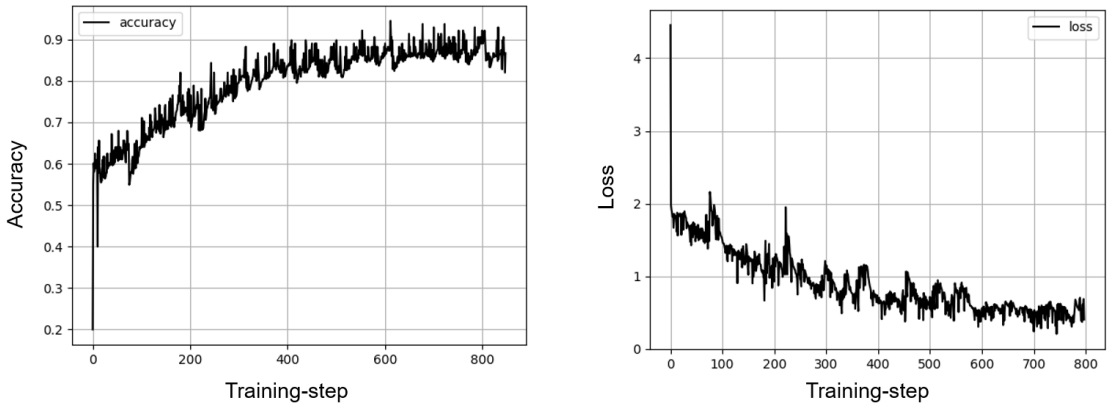
Finally, we calculate the accuracy, recall rate and F1 value on the test set to evaluate the performance of the model in the form of the original dataset. The performance comparison with the previous research work is shown in Table 1 and Table 2. The performance of the baseline model, we refer to the study results of Yang P et al. on the SGM + GE model.

As can be seen from the Figs. 8-a, the accuracy of our model in the validation set in the training process is about 90%. However, since the task of our model is essentially an agent task for multi-step prediction of label sequences, the dataset for training and validation are preprocessed. The higher the number of leading labels, the higher the accuracy of prediction. When testing on the test set, the model needs to start up, that is, gradually predict more and more leading labels. At the beginning prediction of the Figs. 8-a, the accuracy is about 60% which is close to BR model in Table 1. This is because there are no leading labels at beginning prediction, and the model works in a similar fashion to the BR model.

When the error label is used as the leading label, the next prediction will still approximate the startup process after the attention filtering process of the original text. When the model predicts the correct label and uses it as the leading label, the accuracy can reach the level reflected in the training process. Therefore, the performance of our model is different from that on the validation set in the training process when finally predicts in the form of original data on the test set, but the accuracy of 0.802 on AAPD dataset is still better than other models.

D. INTERPRETATION OF RESULT

It can be seen from Figs. 8-b that the validation set loss starts with a rapid vibration decline, then the decline speed tends to moderate and the vibration amplitude gradually decreases, and finally maintains the basic stability. The reason for the



(a) Accuracy in the training process.

(b) Loss in the training process.

FIGURE 8. Training process of our model.

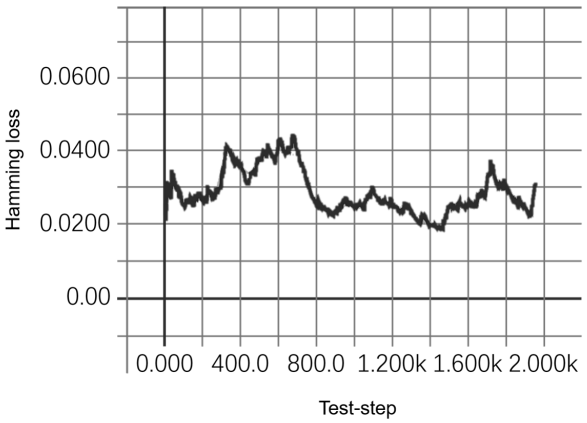


FIGURE 9. Hamming loss in the test process on AAPD dataset.

vibration is the momentum accumulation characteristic of adam optimizer, but the model still shows a macroscopic optimization trend which indicates that the model gradually achieves convergence. At the same time, the accuracy of the model increases and stabilizes with the decrease of loss. The results of our model and other baseline models are shown in Table 1 and Table 2. Our model is superior to other baseline models in terms of accuracy, recall rate and F1 value on two datasets. Meanwhile, Fig. 9 shows that the hamming loss of AAPD dataset remains between 0.02 and 0.04, which is a very low level.

By comparing several typical experimental samples, we demonstrate the ability of our model to accurately predict, avoid repeated predictions and mitigate the accumulation of errors. And the comparison of our model with other baseline models is shown in Table 3. It is clear from this that for a typical sample, our model has better prediction performance than the BR model which ignores the correlation between labels. The improvement of performance comes from the leading label, on the one hand, it provides more comprehensive information for the new labels, on the other hand, it can be measured by the semantic distance of labels and alleviates the problem of sparse data caused by the frequency of label occurrence, and makes labels with lower number get training basis.

From Table 3 we can see that, compared with the attention mechanism model of with encoder and decoder, our model shows high accuracy in the labels of latter part which is embodied in the suppression of error labels. The attention mechanism model loses the leading label in the prediction process of the labels of latter part, and the prediction of the end label becomes difficult. The result shows that the forward propagation of the model results in the loss of the leading label information. Our model provides fuller leading-up label information for labels that appear at the rear. In this case, although our model encountered a startup error, we can see that the model can correct the prediction direction through the attention filtering effect of the original text on the leading label.

As shown in Table 3, our model avoids the extreme situation of continuous false prediction compared with the SGM+GE model. The result indicates that the attention mechanism of the original text information can filter the leading label information as scheduled and the error memory of

TABLE 3. Results example and analysis table. * denotes empty label that the model does not predict.

Real Label	BR	Seq2Seq+Attention	SGM+GM	OurModel(MSCoA)
math.ST	math.ST	math.ST	math.ST	*
cs.AI	*	cs.AI	cs.AI	cs.AI
cs.LG	cs.LG	*	cs.LG	cs.LG
stat.ML	*	stat.ML	*	stat.ML
stat.TH	*	*	*	*
Number of irrelevant labels	3	4	3	2

the leading label caused by one error prediction is eliminated. At the same time, the model avoids the accumulation of errors in several predictions and is helpful to predict the end label in time.

IV. CONCLUSION AND FUTURE WORK

With the development of internet industry, multi-label text classification is becoming more and more widely used than multi-class text classification. Our research work is supported by deep learning and mainly focuses on solving the problem of information loss in the process of sequential decoding. We summarize the theoretical framework of the current research work and improve the performance of multi-label text classification algorithm by making full use of the semantic associations between original text, leading label and new label to be predicted. The experimental results show that our model achieves the optimal level in multi-label text classification task.

Although the experimental results of this study show that our model significantly improves the performance of multi-label text classification tasks, there are still some problems left over from our own point of view. It is difficult to predict the model in the process of startup, that is the model only shows a slightly better performance than the multi-classification model in the case of few leading labels or only startup labels. This means that the model has significant cold start problems. Only when the model begins to have the correct predictive results will it gradually become more and more accurate. This situation is directly related to the mode of work of serialization prediction. How to use the raw data to solve the cold start problem is the next step worth studying.

With the emergence of transformer, bert and other language models, the coding process of multi-label text classification is expected to be greatly improved and broken through. Models can be improved by multi-task learning in future [35]. It is also possible to improve the model effect through transfer learning [36]–[38]. At present, in natural language processing tasks based on word vector, the unsupervised training mode of word vector makes its semantic representation vague and insufficient, and the text representation obtained by coding on this basis cannot fully reflect the semantic content. Specific to the multi-label text classification task, the accurate encoding of the original text information will greatly affect the performance of the model [39]. Therefore, we expect that all kinds of high-performance language models can be applied to the encoder of multi-label text classification tasks in the future.

REFERENCES

[1] C. Shen, J. Jiao, Y. Yang, and B. Wang, "Multi-instance multi-label learning for automatic tag recommendation," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 4910–4914.

[2] S. Gopal and Y. Yang, "Multilabel classification with meta-level features," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 315–322.

[3] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 135–168, May 2000.

[4] M. A. Parwez, M. Abulaish, and J. Jahiruddin, "Multi-label classification of microblogging texts using convolution neural network," *IEEE Access*, vol. 7, pp. 68678–68691, 2019.

[5] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1065–1080, Jun. 2018.

[6] 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.

[7] W. Liu, D. Xu, I. Tsang, and W. Zhang, "Metric learning for multi-output tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 408–422, Feb. 2019.

[8] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5413–5423.

[9] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, "Semantic-unit-based dilated convolution for multi-label text classification," 2018, *arXiv:1808.08561*. [Online]. Available: https://arxiv.org/abs/1808.08561

[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] S. Squartini, A. Hussain, and F. Piazza, "Preprocessing based solution for the vanishing gradient problem in recurrent neural networks," in *Proc. IEEE Int. ISCAS*, vol. 5, May 2003, p. 5.

[12] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," 2015, *arXiv:1509.00685*. [Online]. Available: https://arxiv.org/abs/1509.00685

[13] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.

[14] 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. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[15] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: https://arxiv.org/abs/1508.04025

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[18] S. Sukhbaatar, J. Weston, R. Fergus, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.

[19] W. Zaremba and I. Sutskever, "Reinforcement learning neural turing machines—Revised," 2016, *arXiv:1505.00521*. [Online]. Available: https://arxiv.org/abs/1505.00521

[20] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Adv. Neural Inf. Process. Syst.* Springer, 2001, pp. 42–53.

[21] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 681–687.

[22] 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.

[23] W. Liu and I. W. Tsang, "Large margin metric learning for multi-label prediction," in *Proc. AAAI*, 2015.

[24] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, Dec. 2011.

[25] W. Liu and I. Tsang, "On the optimality of classifier chain for multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 712–720.

[26] W. Liu, I. W. Tsang, and K.-R. Müller, "An easy-to-hard learning paradigm for multiple classes and multiple labels," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3300–3337, 2017.

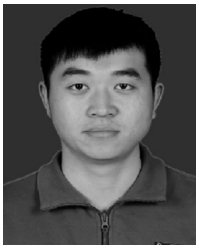
[27] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proc. IEEE IJCNN*, May 2017, pp. 2377–2383.

[28] J. Lin, X. Sun, S. Ma, and Q. Su, "Global encoding for abstractive summarization," in *Proc. ACL*, Jul. 2018, pp. 163–169.

[29] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188

[30] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Proc. ICANN*, 2005, pp. 799–804.

[31] M. Wöllmer, F. Eyben, A. Graves, B. Schuller, and G. Rigoll, "Bidirectional lstm networks for context-sensitive keyword detection in a cognitive virtual agent framework," *Cogn. Comput.*, vol. 2, no. 3, pp. 180–190, 2010.
[32] W. Li, X. Ren, D. Dai, Y. Wu, H. Wang, and X. Sun, "Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions," 2018, *arXiv:1808.05437*. [Online]. Available: https://arxiv.org/abs/1808.05437
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[35] S. Pan, J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Joint structure feature exploration and regularization for multi-task graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 715–728, Mar. 2016.
[36] X. Ben, P. Zhang, Z. Lai, R. Yan, X. Zhai, and W. Meng, "A general tensor representation framework for cross-view gait recognition," *Pattern Recognit.*, vol. 90, pp. 87–98, Jun. 2019.
[37] X. Ben, C. Gong, P. Zhang, R. Yan, Q. Wu, and W. Meng, "Coupled bilinear discriminant projection for cross-view gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, to be published. doi: 10.1109/TCSVT.2019.2893736.
[38] X. Ben, C. Gong, P. Zhang, X. Jia, Q. Wu, and W. Meng, "Coupled patch alignment for matching cross-view gaits," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3142–3157, Jun. 2019.
[39] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.



HAOYANG MA received the B.S. degree in applied physics from the School of Physical Science, Qingdao University, Qingdao, China, in 2017. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Shandong University, Qingdao. His current research interests include machine learning, data mining, multi-label learning, and few-shot learning.

YUJUN LI received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2001. He is currently a Full Professor with the Department of Information Science and Engineering, Shandong University, Qingdao, China. His current research interests include deep learning, natural language processing, multi-label learning, and sentiment analysis.

XIANPENG JI received the B.S. degree in IoT engineering from the School of IoT Engineering, Jiangnan University, Wuxi, China, in 2015. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China. His current research interests include deep learning, natural language processing, and multi-modal sentiment analysis.

JUNLEI HAN received the B.S. degree in electronic information science and technology from the College of Information Science and Technology, Chengdu University of Technology, Chengdu, China, in 2017. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China. His current research interests include deep learning, recommended systems, semantic matching, and natural language processing.

ZEQIANG LI received the B.S. degree in electronic and information engineering from the School of Information Science and Engineering, Shandong University, Jinan, China, in 2017. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Shandong University, Qingdao, China. His current research interests include machine learning, data mining, natural language processing, and sentiment analysis.