Multi-module Fusion Relevance Attention Network for Multi-label Text Classification

Xinmiao Yu, Zhengpeng Li, Member, IANEG, Jiansheng Wu*, and Mingao Liu

***Abstract*—To solve the multi-label text classification (MLTC) task, we propose a multi-module fusion relevance attention network (MFRAN) to explore the semantic correlation between text and category labels. Firstly, the MFRAN model uses a text feature extraction module to capture text information with a strong correlation with category labels and uses multi-head self-attention to obtain the attention score of the corresponding text. Then the learned word-level text semantic information is transmitted to the label attention layer of the category label feature extraction module through multi-dimensional dilated convolution. At the same time, the attention score of category labels is obtained by the bidirectional long short-term memory and label attention layer. The adaptive attention fusion module is used to fuse the text attention score with the attention score of the category label and select the text representation with large output information. We performed a large number of comparative experiments and ablation experiments on the RCV1-V2 and AAPD datasets. The experimental results have proved the MFRAN model is similar to or even exceeds the baseline model when dealing with MLTC tasks.**

***Index Terms*—deep learning, neural network, multi-label text classification, attention mechanism**

I. Introduction

TEXT classification is an important and classic problem in natural language processing (NLP)[1]. In the traditional single-label text classification task, each text sample or instance has only one category label. In single-label classification, each category label is independent of the other and the classification granularity is rough. With the sudden increase of text information, people have a higher and higher degree of granularity of text classification. In the task of multi-label text classification, a text may be associated with multiple category labels, and there is a certain dependence between each category label. The main task of multi-label text classification is to classify a text into multiple labels through a specific classifier or classification network.

With the rapid development of random computers in many fields, multi-label text classification task (MLTC) has been applied in various scenarios, such as news classification[2, 3], sentiment analysis[4, 5], public opinion analysis[6, 7], topic analysis[8, 9], question answering system (QA)[10, 11], information retrieval[12, 13], natural language inference[14, 15], etc. In the face of these extensive downstream applications, multi-label text classification has aroused the interest of researchers. Researchers began to focus on how to extract semantic units containing category labels from text instances, learn the semantic correlation between each text document and its corresponding multiple labels, and fully tap the correlation information such as whether each label has similarities.

For the workflow of multi-label text categorization tasks, like other NLP tasks, the samples are first preprocessed. In the pre-processing stage, the model uses structured data to represent text samples. Text segmentation processing is a key task, through text segmentation technology to extract keywords in the text, including entity words, subject words, etc. The second step is to transform unstructured text information into structured words vector form, such as one-hot coding, a bag of words (BOW) model[16], Word2Vec[17], and Glove[18]. The third step is feature extraction and feature dimension reduction of structured text. After co-occurrence matrix transformation, the feature contained in word vector text is sparse and has a high dimension. How to extract features and reduce dimensions becomes one of the important tasks of the text classification model. Finally, the reduced-dimensional features are sent to the specified classifier for the prediction of category labels. In the validation set and test set, different evaluation indexes (micro-f1, precision, recall, and hamming-loss) are used to evaluate the classification model.

The main contributions of this paper are as follows,

1) A multi-module fusion relevance attention network (MFRAN) is proposed to solve the MLTC task. The MFRAN model pays attention to and learns the semantic correlation between text and category labels, and the semantic correlation between category labels, to improve the accuracy of MLTC.

2) This paper designs two attention mechanism modules and an adaptive attention fusion module. In the text feature extraction module, the multi-head self-attention mechanism is used to capture the word-level semantic information in the text, focusing on semantically relevant category tags. The global information of text attention is transmitted to the label attention layer of the category label feature extraction module by multi-dimensional dilated convolution (MD-Conv). An adaptive attention fusion

Manuscript received March 07, 2022; revised September 02, 2022. The research work was supported by National Natural Science Foundation of China (No.51774179), and Science and Technology Innovation Project of University of Science and Technology Liaoning (LKDYC202109).

Xinmiao Yu is a graduate student of University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 2749936763@qq.com).

Zhengpeng Li is a graduate student of University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 1156361257@qq.com).

Jiansheng Wu* is a professor of University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author, e-mail: ssewu@163.com).

Mingao Liu is a graduate student of University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 1249277436@qq.com)

strategy is designed to fuse the attention of category labels and text and make label predictions.

3) In this paper, a comparative experiment is conducted in the RCV1-V2 dataset and the AAPD dataset. The MFRAN model is analyzed and discussed by four evaluation indexes. Compared with the previous most advanced models, the MFRAN model achieves similar or better performance.

This paper, the rest of the paper is organized as follows. In section 2, the background and related work of the MLTC task is described, and the advantages and disadvantages of each model of MLTC task are analyzed emphatically. Section 3 describes the detailed composition of the MFRAN model. In section 4, two data sets, evaluation indexes, the experimental environment, and model parameters used in the experiment are analyzed. In section 5, we give the experimental results and discuss the results of comparative experiments and ablation experiments. In section 6 a summary of this paper and ideas for future work.

II. BACKGROUND AND RELATED WORK

Since computer development, researchers have tried to use a variety of techniques to solve MLTC. In 2004, Boutell, M.R. et al.[19] proposed using binary relevance (BR) to deal with the problem of classification error when labels overlap in feature space and applied it to semantic scene classification. The BR is the first attempt to convert MLTC problems into multiple single-label problems. However, the BR does not fully consider the correlation between text samples and category labels and ignores the semantic correlation between category labels. In 2007, Tsoumakas, G. and Katakis, I. M.[20] proposed label powerset (LP) to organize sparse related texts into structured representations, trying to fully quantify multiple labels. Read, J.[21] proposed a novel classifier chain (CC) method. CC model is used to model the category labels and achieve the effect of multi-label text classification. However, with the rapid development of computers, the number of samples in the data set has increased dramatically. The problem transformation method taking BR, LP, and CC as examples have the problems of high computational complexity, deep model, and large size. In summary, the problem transformation method has been unable ideally to solve the MLTC problem.

Researchers have been trying to use a convolutional neural network (CNN) to deal with the problem of multi-label text classification since 2014. Wang, P.[22] used word embedding method combined with CNN network to improve the classification accuracy. In 2017, Liu J.[23] also tried to use CNN to extract the correlation information between text and category labels. At the same time, CNN + recurrent neural network (RNN)[24] model was also tried to solve the MLTC problem. The traditional neural network model uses the bag of words (BOW) as the input of the classification model, but this method ignores the context semantic information and deep semantic information. The amount of data used by the MLTC tasks is huge (especially category label types), and large-scale word vector matrix operation consumes large quantities of computing power. The number of networks that are covered by CNN and RNN contributes to the decline in accuracy when it comes to classification. This issue is caused by the model's tendency to forget about the previous moments. To mine the semantic information of text, people try to use a tree structure to divide category labels and make category label predictions. Tree-based models require as much training time as problem transformation methods. In 2018, You, R.[25] proposed AttentionXML. AttentionXML is a representative model of deep learning based on a label tree. AttentionXML has two unique functions. One of the main functions of AttentionXML is its attempt to find the most relevant text information from the various categories of labels. It also proposes a probabilistic label tree that can handle millions of labels. However, the AttentionXML model still has the problem of large size. In 2018, Yang, P.[26] proposed SGM. Different parts of the intercepted text have different contributions to the prediction of different labels. SGM uses the sequence-to-sequence generation model to solve the problem of multi-label text classification.

Obtaining the correlation between category labels has become the research focus of MLTC in recent years. A sample can have multiple category labels, so the research focus of the classification problem is transformed into how to obtain the dependency between labels, thereby improving the accuracy of the classification model. In 2019, Xiao, L.[27] proposed the LSAN model, it uses the semantic information of labels to determine the semantic connection between labels and text documents and constructs the label-specific document representation. A self-attention mechanism is used to identify label-specific document representations from document content information, and an attention fusion strategy is designed to construct a multi-label text classifier. Pal, A.[28] proposed an end-to-end trainable depth network model (MAGNET) combined with a graph attention network in 2020. MAGNET uses a graph neural network to query the semantic relationship between category labels, records the salient features of text by an adjacency matrix, and extracts semantic features at the sentence level by a feature extraction network. However, when this method trains large-scale data, the correlation matrix will be very large, resulting in training difficulties and slow model convergence. Mittal, A.[29] proposed the DECAF algorithm in 2021, and solved the extreme multi-label classification (XML) by learning the classification model composed of rich label metadata. The lightweight text embedding module was used to attach a one-vs-all classifier to each label, this method has better timeliness than the traditional XML method.

Most of these deep learning methods use multiple models to train and predict a large dataset, and most of them use static negative sampling category labels in the training process. Jiang, T., et al. proposed a network model using end-to-end training and dynamic negative label sampling in 2021, LightXML[30], it uses a generative collaboration network to sort and recall text labels. In the recall stage, LightXML recalls both positive and negative label samples, and distinguishes the positive and negative labels in the sorting stage. Although LightXML is a lightweight model, it still performs well in classification accuracy. In recent years, the method based on the deep pre-training model has achieved remarkable results. Wang, Q., et al.[31] found some problems, the pre-training model did not make full use

of the potential space between text samples and category labels. Wang proposed a novel guide network (GUDN) to guide and fine-tune the pre-training model to complete the classification task. The GUDN further improves the prediction accuracy of the model by mining the potential correlation space between text semantic information and category tags.

III. Procedure for Paper Submission

In this section, the formal definition and model overview of the multi-label text classification task is given in section 3.1. In sections 3.2-3.5, each component of the MFRAN model is analyzed and designed in detail.

A. Problem Definition and Model Overview

This paper uses specific mathematical symbols to represent the definition of the MLTC task. It is assumed that $D=\{(x_i,y_i)\}_{i=1}^{n}$ is all the samples in the training set, n represent the total number of samples, and y_i is the category label corresponding to the *ith* sample text instance x_i. X is the text set of all samples, $X=\{x_1,x_2,...,x_i,...,x_n\}$. Y represents the set of category labels corresponding to the sample text, $Y=\{y_1,y_2,...,y_i,...,y_n\}$, $y_i\in\{0,1\}^L$, and L is the total number of class labels. The classification model attempts to find a mapping relationship, $f:X\to Y$. We use D continuous training model to update the mapping relation f. So that it can approximate the real relationship between text and category labels.

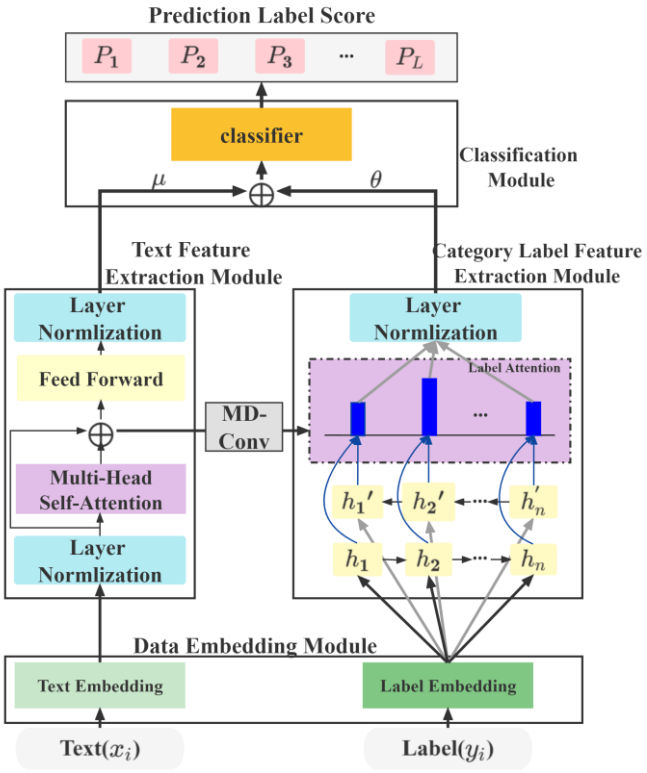

Fig. 1. MFRAN model.

B. Data Representation Module

Independent unstructured text data is not easy to learn, MFRAN model uses the embedding layer to convert the preprocessed samples into the desired vectorized data. The embedding layer (includes 'Text Embedding' and 'Label Embedding'), encodes the text and label of the sample respectively. Text embedding encodes the sample text and transmits it to the text feature extraction module. Label Embedding encodes the sample label and transmits it to the category label feature extraction module.

The text embedding adopts three embedding methods as token embedding, segment embedding, and position embedding. To better learn the context semantic information of text, for the text feature extraction module, we conduct separate pre-training learning for text. Considering the differences in semantic environments of different texts under different data sets, we use the static word vector representation method. The embedding layer is similar to the lookup table, it reduces the word vector to *d*-dimensional space. The embedding layer effectively reduces network computing intensity and training time. The input text is marked before it is sent to the token embedding layer, and two special marks are inserted at the beginning ([CLS]) and ending ([SEP]) of the marked results. Significantly, [CLS] gathers all the features of the text sequence and distinguishes whether the input text has a contextual relationship. [SEP] processes the text by breaking sentences, representing the text token embedding. Segment embedding and position embedding complete the word positioning, they are convenient for the text feature extraction module to learn position information. E_{S}^{T} and E_{P}^{T} represent segment embedding and position embedding respectively. MFRAN model uses the sine function to define position embedding (*PE*). In other words, the vector determines the relative distance between different tokens in a sentence. The *PE* formula is as follows,

$$PE\left(pos_n,2\times ind\right)=\sin\left(\frac{pos_n}{10000^{\frac{2\times ind}{d}}}\right), \tag{1}$$

$$PE\left(pos_n,2\times ind+1\right)=\cos\left(\frac{pos_n}{10000^{\frac{2\times ind}{d}}}\right), \tag{2}$$

where $pos_n\in\mathbb{R}^{Len\times d}$ refers to the absolute position of the nth word in the original sentence, *d* represents the dimension coefficient of embedding. *Len* is the maximum length of position information and is also the maximum length of the input text. *ind* is the index of each value in the pointer. When sine coding is used in the even position, sine coding is used in the odd position.

The output of the text embedding layer E_i^{Text} is also the input word vector matrix of the text feature extraction module and E_i^{Text} is defined as follows,

$$E_i^{Text}=E_{\mathrm{t}}^{T}\oplus E_{\mathrm{S}}^{T}\oplus E_{\mathrm{P}}^{T}, \tag{3}$$

where $\oplus$ represents matrix addition operations.

The label embedding layer only uses token embedding to convert the class label of the text, and the class label after embedding is expressed as E_i^{Label}. E_i^{Label} is the output of the label embedding layer, it is also the input of the category label feature extraction module.

C. Text Feature Extraction Module

For MLTC tasks, the MFRAN model uses a multi-head self-focus mechanism to implement different degrees of attention for different words (including the same word but in different locations) to calculate the representation of sequences. The model also retains the text hidden vector H^{Text} . This method is similar to the training mechanism of the large-scale pre-training model. Finally, multiple attention spaces are connected to the residual of E_i^{Text} after layer normalization to obtain the attention representation of text A^T .

The attention mechanism is essentially an addressing process. The model gives a task-related query vector (*Q*) and calculates attention value by calculating the attention distribution with key (*K*) and attaching it to value (*V*). The calculation formula of attention is as follows,

$$Attention(Q,K,V) = SoftMax(\frac{(QK^T)}{\sqrt{(d_k)}})V, \quad (4)$$

where $\sqrt{(d_k)}$ is a regulating control coefficient. Multi-head self-attention has multiple independent attention spaces, with a different query, key, and value weight matrices W^Q, W^K, W^V in each attention. Each attention mechanism parameter in the 12-head self-attention mechanism is not shared. The *ith* attention represents the subspace is $Space_i$.

$$Space_i = Attention\left(Q_i W^{(Q_i)}, K_i W^{(K_i)}, V_i W^{(V_i)}\right), \quad (5)$$

$$A^T = Concat(Space_1, Space_2, ..., Space_i, ..., Space_{12}) \oplus LN(E_i^{Text}), \quad (6)$$

where $LN(E_i^{Text})$ is the word vector matrix after layer normalization of E_i^{Text} .

In the text feature extraction module, we try to focus on the semantic information in the text content. It is not enough to select semantic information based on a multi-headed self-attention mechanism only considering the impact of the text itself on category labels. The text feature extraction module strengthens the part of text representation by the feed-forward layer and layer normalization layer.

The feed-forward layer is essentially a two-layer full-connection layer. On the internal structure of the multi-head self-attention layer, the MFRAN model primarily performs the point-product attention of scaling, that is to say the linear transformation of the text word vectors. The learning capacity of a linear transformation is less valuable than a non-linear transformation. Although the outcome of multi-head self-attention uses the attention mechanism to learn the new representation of each word, the expressive capacity of this representation may not be strong. For this reason too, a normalization layer is added after the attention layer. By standardizing representation, the standardized word vector is moved to the action area of the activation function, it can make the *Relu* activation function play a better role. At the same time, in the full-connection layer, the model can learn more abstract text semantic information by mapping the word vector matrix to the high-dimensional space and then to the low-dimensional space. The feedforward layer makes the expression ability of word representation stronger, and more able to represent the relationship between words and other words in context. The attention of the text feature extraction module to the text is expressed as A^{Text} , are defined as follows,

$$A^{Text} = Relu(W_1 \times A^T + b_1)W_2 + b_2. \quad (7)$$

D. Category Label Feature Extraction Module

Category labels also contain semantic information in multi-label text classification problems. Independently using multi-head self-attention to focus on semantic information at the word level may violate the semantic information of the source label of the text. We use the category label feature extraction module and classification module to correct this problem. MFRAN model attempts to learn the semantic information of the text content and the category label of the sample respectively.

To avoid the long-range dependence problem in the traditional CNN network, the bidirectional long short term memory (Bi-LSTM)[32] is used as the backbone network model of category labels. The label embedding layer generates the corresponding label representation E_i^{Label} as to the input, and Bi-LSTM contains both forward and backward propagation. Therefore, the model can pay attention to the semantic association information between each category label, greatly alleviate the long-range dependence problem, and avoid the situation the new features learned from the model will cover the old features. Bi-LSTM uses the same three gate control unit components as LSTM: input gate, output gate and forget gate. Bi-LSTM also considers the semantic information before and after the moment and updates the hidden layer state at each moment of the model through the word embedding of text documents and the word embedding of category labels. The hidden layer state at the *i*th moment can be expressed as,

$$\overrightarrow{h_i^{Label}} = \overrightarrow{LSTM}(\overrightarrow{h_{i-1}^{Label}}, y_i), \quad (8)$$

$$\overleftarrow{h_i^{Label}} = \overleftarrow{LSTM}(\overleftarrow{h_{i-1}^{Label}}, y_i), \quad (9)$$

where y_i is the embedded vector representation of the *i*th category label. The final hidden representation of the *i*th word h_i^{Label} is obtained by combining the hidden states of the two directions.

$$h_i^{Label} = [\overrightarrow{h_i^{Label}}, \overleftarrow{h_i^{Label}}], \quad (10)$$

where $\overrightarrow{h_i^{Label}}, \overleftarrow{h_i^{Label}} \in \mathbb{R}^k$, $\overrightarrow{h_i^{Label}}$ represents the forward word context of the Bi-LSTM model and $\overleftarrow{h_i^{Label}}$ represents the backward context. Then the hidden layer of the entire category label is inferred to be $\overleftarrow{H^{Label}} \in \mathbb{R}^{2k \times n}$,

$$\overrightarrow{H^{Label}} = [\overrightarrow{h_1^{Label}}, \overrightarrow{h_2^{Label}}, ..., \overrightarrow{h_{i-1}^{Label}}, \overrightarrow{h_i^{Label}}], \quad (11)$$

$$\overleftarrow{H^{Label}} = [\overleftarrow{h_1^{Label}}, \overleftarrow{h_2^{Label}}, ..., \overleftarrow{h_{i-1}^{Label}}, \overleftarrow{h_i^{Label}}], \quad (12)$$

$$\overleftarrow{H^{Label}} = [\overrightarrow{H^{Label}}, \overleftarrow{H^{Label}}], \quad (13)$$

MD-Conv generates text semantic units through local correlation and long-term dependence between texts. MFRAN model focuses on word-level text semantic information through the MD-Conv layer and sets a small dilation rate to avoid the influence of long-distance information on the semantic unit.

MFRAN model transfers the text attention representation A^T through an MD-Conv layer to the label attention layer

of the category label feature extraction module and combines it with the hidden layer representation $\overleftarrow{H^{Label}}$. MD-Conv generates text semantic units through local correlation and long-term dependence between texts. MFRAN model focuses on word-level text semantic information through MD-Conv, thereby increasing the receptive field of text semantic information. Adjusting the dilation rate parameter allows the model to learn more semantic correlations between text and category labels, and a smaller dilation rate can avoid the influence of long-distance information on semantic units. CNN can only focus on near semantic information and has a strong ability to capture local dependencies, but it will lose long-distance text semantic information. A^{Text} transmits the semantic information unit of text through MD-Conv, enhances the semantic information correlation between text and category labels, effectively solves the problem of no correlation between remote semantic information, and prevents the loss of local information and remote information.

The word vector matrix E_i^{Label} of the category label is combined with the hidden layer state $\overleftarrow{H^{Label}}$ by matrix multiplication, and the MD-Conv is used to supplement the semantic relationship between the text and the category label. Finally, the category label feature extraction module is obtained for the category label attention representation A^{Label},

$$\overrightarrow{A^{Label}} = E_i^{Label} \bullet \overrightarrow{H^{Label}} \oplus MDC(A^T) \bullet \overrightarrow{H^{Label}}, \quad (14)$$

$$\overleftarrow{A^{Label}} = E_i^{Label} \bullet \overleftarrow{H^{Label}} \oplus MDC(A^T) \bullet \overleftarrow{H^{Label}}, \quad (15)$$

$$A^{Label} = [\overrightarrow{A^{Label}}, \overleftarrow{A^{Label}}], \quad (16)$$

where *MDC* is the output matrix of the MD-Conv layer.

E. Classification Module

Through the text feature extraction module and category label feature extraction module, the attention representation of text A^{Text}, the hidden layer of text H^{Text}, the attention score of category label A^{Label}, and the hidden layer of category label $\overleftarrow{H^{Label}}$ are obtained respectively. Then the two attention mechanisms are weighted fusion by the classification module. We can deduce the final attention score A^{out}.

$$\mu = sigmoid(W_\mu \bullet A^{Text} \bullet H^{Text}), \quad (17)$$

$$\theta = sigmoid(W_\theta \bullet A^{Label} \bullet \overleftarrow{H^{Label}}), \quad (18)$$

$$A^{out} = \mu \bullet A^{Text} \oplus \theta \bullet A^{Label}, \quad (19)$$

where W_μ, W_θ are the training parameter, and μ, θ are the attention weight vector.

Finally, the text classifier is constructed by a multi-layer perceptron, and the sigmoid function is used to convert the output value to the category label score in the (0,1) interval, and a threshold is set to complete the multi-label text classification task.

MFRAN model uses cross-entropy loss as the loss function to calculate the loss value $\mathbb{L}$,

$$\mathbb{L} = -\sum_{i=1}^{N}\sum_{j=1}^{L}(y_{ij}\log(y_{ij})) + (1-y_{ij})\log(1-y_{ij}) \quad (20)$$

where N is the number of training documents, L is the number of labels, $y_{ij} \in [0,1]$ is the predicted score, and y_{ij} indicates the ground truth of the *ith* text along with the *jth* label.

IV. Datasets and Implementation Details

In this section, we evaluate our proposed methods on two datasets. We first introduce the datasets, evaluation metrics, experimental details, and all baselines. Then, we compare our methods with the baselines. Finally, we provide the analysis and discussions of experimental results.

A. DataSets

We use two public datasets to verify the performance of the MFRAN model, and the statistics of these two datasets are shown in table 1.

TABLE 1.
EXPERIMENTAL DATASET.
Sample numbers represent the total amount of sample data contained in the dataset. Label numbers refer to the number of class labels in the dataset. Sample word average is the average number of words in each sample text. Sample label average is the average number of class labels the sample of belonging to. Maximum number of labels represents the maximum number of labels for a single sample in the dataset.

Dataset	Sample numbers	Label numbers	Sample word average	Sample label average	A maximum number of labels
RCV1-V2	804414	103	123.9	3.2	17
AAPD	55840	54	163.4	2.4	12

RCV1-V2: RCV1-V2 is a large-scale text data set proposed by Lewis, D. et al.[33] in 2004. It contains a total of 804,414 news datasets, each of which has multiple class labels, and a total of 103 class labels (topics), each of which is represented by a string. Their label frequencies span five orders of magnitude, from 5 times 'GMIL' to 381327 times 'CCAT'. RCV1-V2 was split in chronological order: the first 23149 samples served as a training set and the last 781265 samples served as a test set. In this paper, RCV1-V2 is randomly divided into the training set, validation set, and test set according to the ratio of 6:2:2.

Arxiv Academic Paper Dataset (AAPD): Compared with RCV1-V2, AAPD[34] is a small dataset with only 55840 data samples. All the data samples in AAPD are from the arxiv database. All the samples are research papers in the computer field, and the content may be the summary part or other part of the computer papers. AAPD dataset has 54 category labels, and we also randomly split the AAPD dataset according to the ratio of 6:2:2.

B. Implementation Details

We use a 3080ti series graphics card to complete the classification experiment, detailed experimental environment reference table 2.

The parameter settings of the MFRAN model in the RCV1-V2 dataset are as follows,

1) We use the word table size of 50,000. The batch size of the MFRAN model is set to 8, the training round epoch is set to 20, and the maximum length of document content data input max-len is set to 512.

2) The encoding dimension of the text feature extraction module is 768, and the multi-head self-attention mechanism has 12 heads.

3) Using the Adam[35] loss function (Adaptive Moment Estimation) with β1 = 0.8, β2 = 0.9, class-weight is used before the loss function to mitigate the problem of the loss function had insufficient attention to samples with less data because of unequal datasets.

4) Initial learning rate lr＝le-5. If there is no learning change in the two-batch, the learning rate will be halved $lr_{new} = \frac{1}{2} lr_{previoustime}$.

TABLE 2.
EXPERIMENTAL ENVIRONMENT

Experimental environment	Experimental configuration
Operating system	Ubuntu18.04
Programming language	Python3.6
Deep Learning Framework	Pytorch1.1/Pytorch1.4
Display card model	NVIDIA GeForce GTX3080ti (12G)

The parameter settings of the MFRAN model in the AAPD dataset are roughly the same as those in the RCV1-V2 dataset. We only make corresponding adjustments in the model training vocabulary and model training batches. On the AAPD dataset, MFRAN model training round epoch = 15, and vocabulary = 30000.

C. Baseline Model

The baseline models used in this paper include:

1) **Binary Relevance (BR):** Boutell, M. R., et al.[19] proposed in 2004 to use binary relevance (BR) to deal with the problem of classification error when categories overlap in feature space and applied it to semantic scene classification. BR first attempted to convert the MLTC problem into multiple single-label classification problems.

2) **Label Powerset (LP):** Tsoumakas, G. and Katakis, I. M.[20] proposed a problem transformation method to solve MLTC tasks in 2007. LP method is similar to BR, and CC, however, with the development of the neural network model this transformation method has been gradually banned.

3) **Classifier Chains (CC):** Read, J., et al.[21] proposed in 2011 to treat each label of MLTC as an independent binary problem, taking into account the lack of awareness of label correlation modeling. CC uses a novel classifier chain method to model the label correlation while maintaining an acceptable computational complexity.

4) **CNN:** In 2017, Liu, J., et al.[23] first attempted to apply deep learning to XMTC, and proposed a set of CNN models. The text features in training samples are extracted by continuous convolution operation, and the distribution probability matrix of class labels is output by sigmoid after the full connection layer.

5) **CNN-RNN:** Chen, G., et al.[24] proposed an integrated application of RNN and CNN in 2017 to capture global and local text semantics. The category label is modeled and the label correlation is analyzed in the case of computable complexity. This method solves the long-range dependence problem caused by using the CNN network alone and improves the classification accuracy of MLTC by integrating networks.

6) **SGM:** Yang, P., et al.[26] proposed a sequence generation model (SGM) for multi-label classification in 2018, it treats the multi-label classification task as a sequence generation problem to model the correlation between labels, and applies a sequence generation model with a novel decoder structure to solve the MLTC task.

7) **Seq2Set:** Yang, P., et al.[36] proposed a simple but effective sequence-to-set model (Seq2Set) in 2019. Seq2Set is trained by reinforcement learning. Seq2Set contains a reward feedback mechanism to learn the order of category labels. In this way, Seq2Set reduces the model's dependence on label order and captures high-order correlations between labels.

8) **LSAN:** Xiao, L., et al.[27] proposed a label-specific attention network (LSAN) in 2019. LSAN builds a semantic connection between the category label and the training document by fully referring to the semantic information of the label. Transform the semantic expression of the document to make it more suitable for document labels.

9) **MAGNET:** Pal, A., et al.[28] proposed a graph-based attention network model in 2020 to capture the attention dependence between labels. MAGNET is an end-to-end model. MAGNET captures and explores key dependencies between labels using feature matrices and correlation matrices and classifies sentence feature vectors obtained from the text.

D. Evaluation

Considering the MFRAN model mainly solves the MLTC task, the RCV1-V2 dataset and the AAPD dataset used in this paper have the problem of unbalanced sample data, especially the long tail distribution of the RCV1-V2 dataset. To comprehensively and objectively evaluate the multi-classification effect of the model designed in this paper, the Micro-F1 score and Hamming-loss are used as the main evaluation indexes of this model.

Micro-F1 ($F1_{micro}$) is the harmonic mean of precision (P) and recall (R). The larger the F1-Score is, the higher the classification accuracy of the model is, and vice versa. Firstly, the total precision and recall of all categories are calculated, and then the Micro-F1 value is calculated by using the formula. The detailed formula is as follows,

$$Precision_{micro} = \frac{TP}{TP + FP}, \tag{21}$$

$$Recall_{micro} = \frac{TP}{TP + FN}, \tag{22}$$

$$F1_{micro} = 2 \times \frac{Precision_{micro} \times Recall_{micro}}{Precision_{micro} + Recall_{micro}}, \tag{23}$$

Hamming-loss is a method for obtaining accuracy in multi-label classification scenarios. *Hamming-loss* calculates the average accuracy of all samples.

For each classified text sample, the accuracy rate is the proportion of the predicted correct number of labels in the total predicted correct number and the true correct number of labels. The smaller the *Hamming-loss* is, the better the classification effect of the model is. The *Hamming-loss* is as follows,

$$Hamming\text{-}Loss = \frac{1}{n} \sum_{i=1}^{n} \frac{XOR(y_{ij}, y_{ij})}{L}, \tag{24}$$

where n is the total number of samples, L is the number of

labels, y_{ij} is the *jth* real label corresponding to the *ith* sample, y_{ij} is the *jth* prediction label corresponding to the *ith* sample, and *XOR*(•) is the XOR operation.

V. Results and Discussion

To verify the effectiveness of the MFRAN model on MLTC tasks and improve the authenticity and reliability of the experimental results, we use the same hyperparameters and experimental configuration environment. Ten experiments were conducted on the RCV1-V2 dataset and AAPD dataset, and the average value of the experimental results was taken as the final experimental result. The comparative experimental results are shown in tables 3–4. The experimental results of all baseline models in this paper are either derived from the original paper or retrained according to the parameters of the paper.

For the RCV1-V2 dataset, the MFRAN model achieved 88.0% (0.880) in indicators, and the MFRAN model reached 70.7% (0.707) on the AAPD dataset.

Compared with the BR model[19], LP model[20], and CC model[21] solve the MLTC task by problem transformation method, the MFRAN model designed in this paper has about 3.5%(0.035) improvement.

By observing tables 3-4, it is found the time and space occupied by the method of converting the multi-classification problem into binary is huge when dealing with large-scale data sets (RCV1-V2 dataset), and most of them directly ignore the semantic correlation between category labels and do not fully mine the correlation between text and category labels, resulting in low *P*, *R*, and *F*1 indexes. Especially in the case of less text data content, the deficiency of this method is more obvious.

The MFRAN model captures the hidden semantic information between category labels through the Bi-LSTM model, learns the semantic information of word-level text through multi-head self-attention, and effectively mines the semantic correlation between text and category labels through the fusion of classification module text and label attention.

Tabel 3.
The results of different algorithms on RCV1-V2

Models	Datasets	P(%)	R(%)	F1(%)
BR	RCV1-V2	90.4	81.6	85.8
LP		89.6	82.4	85.8
CC		88.7	82.8	85.7
CNN		**92.2**	79.8	85.5
CNN-RNN		88.9	82.5	85.6
SGM		88.7	85.0	86.9
Seq2Set		90.0	**85.8**	87.9
LSAN		91.3	84.1	87.5
MAGNET		-	-	-
MFRAN		91.8	84.4	**88.0**

In lines 4-5, we find the CNN model performs better than the MFRAN model on precision (*P*), however, their low scores lead to poor Micro-F1 scores. Some CNN models do not perform better than the problem transformation method on recall (*R*). CNN model has strict requirements for the input format, and cannot process data in the form of sequence. Most of the input in natural language processing is sequence data, for deep models like CNN and RNN, with the increase of network layer, there will be obvious gradient disappearance and gradient explosion problems. In dealing with nonlinear problems, the CNN[22, 23] or CNN+RNN[24] models have poor classification results. Researchers use the network with an improved RNN structure to solve the long-range dependence problem, as the Bi-LSTM model used by the MFRAN model is representative of the improved RNN model.

Tabel 4.
The results of different algorithms on AAPD

Models	Datasets	P(%)	R(%)	F1(%)
BR	AAPD	64.4	64.8	64.6
LP		65.7	65.1	63.4
CC		65.7	65.1	65.4
CNN		**84.9**	54.5	66.4
CNN-RNN		71.8	61.8	66.4
SGM		74.6	65.9	69.9
Seq2Set		73.9	**67.4**	70.5
LSAN		77.7	64.6	70.6
MAGNET		-	-	69.6
MFRAN		78.1	64.7	**70.7**

MFRAN model uses multi-head self-attention and multi-label attention to capture the semantic correlation between rational labels and texts. This method has been significantly improved in the extraction of text semantic information and the reduction of model training time. However, this deep learning method still has shortcomings. They are far less interpretable than machine learning.

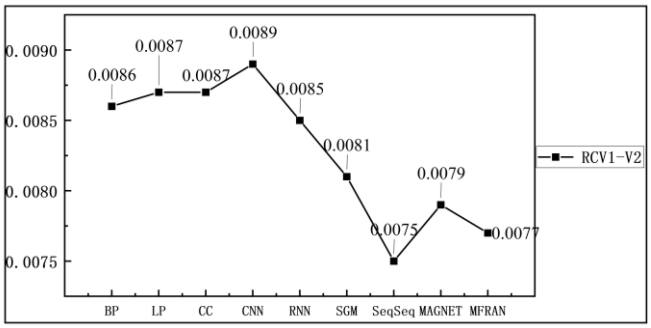

Fig. 2. Hamming-loss of MFRAN model on the RCV1-V2 dataset.

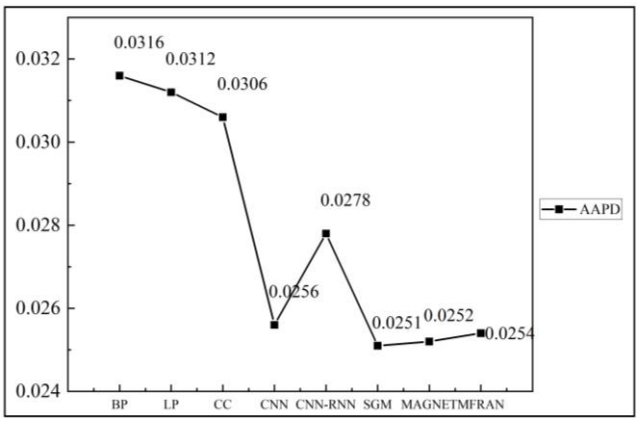

Fig. 3. Hamming-loss of MFRAN model on the AAPD dataset.

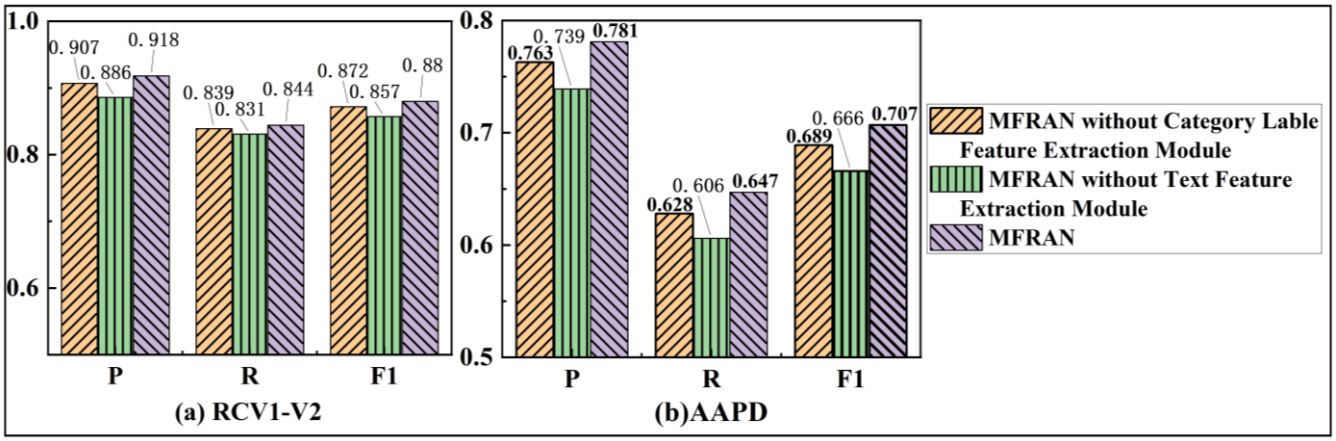

Fig.4. Ablation experiments of MFRAN model.

We compare the last five rows of tables 3-4. The MFRAN model is basically flat or even better than the baseline model on precision, recall, and *F*1. This also fully proves the effectiveness of the MFRAN model in the multi-classification problem. To further verify the number of MFRAN model misclassification labels (labels not belonging to this sample are predicted, or labels belonging to this sample are not predicted), we calculate Hamming-loss scores on the RCV1-V2 dataset and AAPD dataset, respectively, as shown in figure 2 and figure 3. We find the MFRAN model still needs to be improved in dealing with the problem of unbalanced sample data, and the model error prediction is relatively more on the AAPD dataset. Compared with RCV1-V2, AAPD belongs to a small-scale dataset. When dealing with MLTC tasks, it faces 54 categories of labels, the text data of some labels are less and the semantic correlation between each label is strong.

Although the category labels of the RCV1-V2 dataset are much more than those of the AAPD dataset (about twice), the RCV1-V2 dataset has a large database (about 14.5 times), and the smallest category labels still have more data than the AAPD dataset. The *Hamming-loss* score of the AAPD dataset is not as good as SGM[26] or MAGNET[28], roughly the same, compared with the traditional neural network model has been significantly improved.

The MFRAN model proposes a multi-module fusion relevance attention network to capture semantic information between text and category labels, MFRAN has three attention modules, a text feature extraction module, a category label feature extraction module, and a classification module. To verify the necessity of each module component, we conducted the ablation comparative experiment, the experimental results are shown in figure 4. The 'MFRAN with Category Lable Feature Extraction Module' represents only using the text feature extraction module to learn the semantic information of the text. The 'MFRAN with Text Feature Extraction Module' represents only using the category label feature extraction module to learn the semantic information between category labels. The 'MFRAN' represents both learning the semantic information of text and category labels and fuses the two attention by classification module. Figure 4 shows the score of the 'MFRAN' method on the RCV1-V2 dataset being significantly higher than others alone. This situation is also confirmed in the AAPD dataset. It is reasonable and effective to consider the correlation between text and category labels, and the correlation between labels and labels.

VI. CONCLUSION

In the face of a multi-label text classification task, this paper proposes a multi-module fusion relevance attention network for multi-label text classification (MFRAN). MFRAN completed the MLTC task through two attention mechanism modules and an adaptive attention fusion module. The multi-head self-attention was used to complete the learning of text content, Bi-LSTM and label attention were used to obtain the hidden semantic information of labels, and the adaptive attention fusion module was used to realize the weighted fusion of text attention and category label attention, to effectively capture the semantic correlation between text and category label. We conducted a large number of comparative experiments on two open-source datasets, and the experimental results show the MFRAN model being superior to most existing multi-label classification models. In the future work, we will consider how to effectively solve the extreme multi-label text classification task, especially when the long tail distribution of the data set is obvious, and how to ensure the classification accuracy of the model.

REFERENCES

[1]Z. Li, J. Wu, J. Miao, X. Yu, and S. Li, "Multi-model Fusion Attention Network for News Text Classification," *International Journal for Engineering Modelling*, vol. 35, no. 2, pp. 1-15, 2022.

[2]Y. Hu, X. Zhang, J. Yang, and S. Fu, "A Hybrid Convolutional Neural Network Model Based on Different Evolution for Medical Image Classification," *Engineering Letters*, vol. 30, no. 1, pp. 168-177, 2022.

[3]S. Arwatchananukul, R. Saengrayap, S. Chaiwong, and N. Aunsri, "Fast and Efficient Cavendish Banana Grade Classification using Random Forest Classifier with Synthetic Minority Oversampling Technique," *IAENG International Journal of Computer Science*, vol. 49, no. 1, pp. 46-54, 2022.

[4]C. Henriquez and G. Sanchez-Torres, "Aspect extraction for opinion mining with a semantic model," *Engineering Letters*, vol. 29, no. 1, pp. 61-67, 2021.

[5]A. A. Syed, F. L. Gaol, W. Suparta, E. Abdurachman, A. Trisetyarso, and T. Matsuo, "Prediction of the Impact of Covid-19 Vaccine on Public Health Using Twitter," *IAENG International Journal of Computer Science*, vol. 49, no. 1, pp. 19-29, 2022.

[6]G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, pp. 51522-51532, 2019.

[7]J. Tang and X. Xu, "Research on Detection of Chinese Microblog Public Opinion Analysis System," in *11th International Conference on*

Computer Engineering and Networks, CENet2021, October 21, 2021 - October 25, 2021, Hechi, China, 2022, vol. 808 LNEE: Springer Science and Business Media Deutschland GmbH, pp. 766-773.

[8] J. Feng, X. Mu, W. Wang, and Y. Xu, "A topic analysis method based on a three-dimensional strategic diagram," *Journal of Information Science*, vol. 47, no. 6, pp. 770-782, 2021.

[9] R. Setiawan, Salmah, Widodo, I. Endrayanto, and Indarsih, "Analysis of the Single-Vendor-Multi-Buyer Inventory Model for Imperfect Quality with Controllable Lead Time," *IAENG International Journal of Applied Mathematics*, vol. 51, no. 3, pp. 645-654, 2021.

[10] E. Rakovska and M. Hudec, "A Three-Level Aggregation Model for Evaluating Software Usability by Fuzzy Logic," *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 3, pp. 489-501, 2019.

[11] J. Lodhavia, V. Jain, A. Gupta, H. Prabhu, and A. Pandit, "User-profiling based E-learning system using Question Adaptation," in *2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020, July 15, 2020 - July 17, 2020*, Coimbatore, India, 2020: Institute of Electrical and Electronics Engineers Inc., pp. 723-728.

[12] Z. Chen, Z. Wang, and S. A. Jafar, "The Capacity of T-Private Information Retrieval with Private Side Information," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4761-4773, 2020.

[13] Y. Chaudhary, H. Schutze, and P. Gupta, "Explainable and discourse topic-aware neural language understanding," in *37th International Conference on Machine Learning, ICML 2020, July 13, 2020 - July 18, 2020*, Virtual, Online, 2020, vol. PartF168147-2: International Machine Learning Society (IMLS), pp. 1456-1465.

[14] T. Schick and H. Schutze, "Exploiting cloze questions for few shot text classification and natural language inference," in *16th Conference of the European Chapter of the Associationfor Computational Linguistics, EACL 2021, April 19, 2021 - April 23, 2021*, Virtual, Online, 2021: Association for Computational Linguistics (ACL), pp. 255-269.

[15] W. Quamer, P. K. Jain, A. Rai, V. Saravanan, R. Pamula, and C. Kumar, "SACNN: Self-attentive Convolutional Neural Network Model for Natural Language Inference," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 3, 2021.

[16] L. S. Shankar, A. Sravani, T. S. Kumar, S. Rajender, and C. M. A. K. Z. Basha, "Convolution Neural Network (CNN) Based Computerized Classification of Adulterated Fruits with SIFT and Bag of words (BOW)," in *4th International Conference on Smart Systems and Inventive Technology, ICSSIT 2022, January 20, 2022 - January 22, 2022*, Tirunelveli, India, 2022: Institute of Electrical and Electronics Engineers Inc., pp. 1068-1073.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, May 2, 2013 - May 4, 2013*, Scottsdale, AZ, United states, 2013: International Conference on Learning Representations, ICLR.

[18] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25, 2014 - October 29, 2014*, Doha, Qatar, 2014: Association for Computational Linguistics (ACL), pp. 1532-1543.

[19] 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.

[20] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," 2007, vol. 3: IGI Publishing, pp. 1-13.

[21] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333-359, 2011.

[22] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. 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.

[23] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017, August 7, 2017 - August 11, 2017*, Tokyo, Shinjuku, Japan, 2017: Association for Computing Machinery, pp. 115-124.

[24] 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 2017, May 14, 2017 - May 19, 2017*, Anchorage, AK, United states, 2017, vol. 2017-May: Institute of Electrical and Electronics Engineers Inc., pp. 2377-2383.

[25] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, and S. Zhu, "AttentionXML: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification," in *33rd Annual Conference on Neural Information Processing Systems, NeurIPS 2019, December 8, 2019 - December 14, 2019*, Vancouver, BC, Canada, 2019, vol. 32: Neural information processing systems foundation, p. Citadel; Doc.AI; et al.; Lambda; Lyft; Microsoft Research.

[26] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *27th International Conference on Computational Linguistics, COLING 2018, August 20, 2018 - August 26, 2018*, Santa Fe, NM, United states, 2018: Association for Computational Linguistics (ACL), pp. 3915-3926.

[27] L. Xiao, X. Huang, B. Chen, and L. Jing, "Label-specific document representation for multi-label text classification," in *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, November 3, 2019 - November 7, 2019*, Hong Kong, China, 2019: Association for Computational Linguistics, pp. 466-475.

[28] A. Pal, M. Selvakumar, and M. Sankarasubbu, "Magnet: Multi-label text classification using attention-based graph neural network," in *12th International Conference on Agents and Artificial Intelligence, ICAART 2020, February 22, 2020 - February 24, 2020*, Valletta, Malta, 2020, vol. 2: SciTePress, pp. 494-505.

[29] A. Mittal *et al.*, "DECAF: Deep Extreme Classification with Label Features," in *14th ACM International Conference on Web Search and Data Mining, WSDM 2021, March 8, 2021 - March 12, 2021*, Virtual, Online, Israel, 2021: Association for Computing Machinery, Inc, pp. 49-57.

[30] T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, and F. Zhuang, "LightXML: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification," arXiv, 2021.

[31] Q. Wang, H. Shu, and J. Zhu, "GUDN: A novel guide network for extreme multi-label text classification," arXiv, 2022.

[32] J. Li, Y. Xu, and H. Shi, "Bidirectional LSTM with Hierarchical Attention for Text Classification," in *4th IEEE Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2019, December 20, 2019 - December 22, 2019*, Chengdu, China, 2019: Institute of Electrical and Electronics Engineers Inc., pp. 456-459.

[33] 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.

[34] P. Yang, X. Sun, W. Li, and S. Ma, "Automatic academic paper rating based on modularized hierarchical convolutional neural network," in *56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, July 15, 2018 - July 20, 2018*, Melbourne, VIC, Australia, 2018, vol. 2: Association for Computational Linguistics (ACL), pp. 496-502.

[35] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, May 7, 2015 - May 9, 2015*, San Diego, CA, United states, 2015: International Conference on Learning Representations, ICLR.

[36] P. Yang, F. Luo, S. Ma, J. Lin, and X. Sun, "A deep reinforced sequence-to-set model for multi-label classification," in *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019, July 28, 2019 - August 2, 2019*, Florence, Italy, 2020: Association for Computational Linguistics (ACL), pp. 5252-5258.