Mining Knowledge within Categories in Global and Local Fashion for Multi-Label Text Classification

Sheng Bi, Peng Shi, Yuntao Du, Bin Jin, Lingshuang Yu
National Key Laboratory for Novel Software Technology, Nanjing University
Department of Computer Science and Technology
Nanjing, China
benson.bi.cs@gmail.com,spwannasing@gmail.com,dz1833005@smail.nju.edu.cn,
kimbin@smail.nju.edu.cn,yu.lingshuang@outlook.com

***Abstract*—Multi-label text classification (MLTC) is an important task in natural language processing, which assigns multiple labels to each text in the dataset. Typical method like Binary Relevance (BR) is arguably the most intuitive solution for the task. It works by decomposing the multi-label learning task into a number of independent binary learning tasks while ignoring the correlation between labels. Recently, neural network models attract much attention. Researchers view the MLTC task as a sequence generation problem. Although some new methods based on generative model (e.g. sequence-to-sequence), such as novel decoder structure and various attention mechanisms, can improve the performance. These methods still have some shortcomings, such as unreasonable loss function, unclear ordering of target labels. To address these limitations, we propose a simple and effective novel model, which combines the merits of neural network and BRs methods. Our model also takes into account the categories and levels of labels. We decompose the MLTC problem to binary classification, together with global and local extractor to avoid the impact of label ordering and cumulative error. Experimental results show that our model achieves an improvement of 3.0% micro-F1 and a reduction of 6.0% hamming loss on AAPD dataset compared with the state-of-the-art work. And obtained good performance on RCV1-V2 dataset.**

***Index Terms*—Multi-label, Nerual Networks, Global-and-Local Extractor**

I. Introduction

Multi-label text classification (MLTC) is an important task in natural language processing (NLP), which assigns multiple labels to each text in the dataset. MLTC enables a broad range of applications, and one common real-world scene is the news' labels classification. In order to accurately and effectively recommend the news to its right users, we need to classify the news. Usually, the content of news involves multiple topics (labels), and this is where multi-label classification comes in.

To tackle this task, many efficient methods had been proposed. Binary Relevance (BR) [33] transforms the MLTC into multiple single-label classification problem. Specifically, BR procedure works in an independent manner, where the binary classifier for each class label is learned by ignoring the existence of other class labels [22]. Hence, many correlation-enabling extensions to binary relevance have been proposed [19], [23], [24], [35]. Recently, neural network models have made a remarkable achievement in NLP. Inspired by the tremendous success of the sequence-to-sequence (seq2seq) model, there are many innovations [31], [41], [43] based on it, such as novel decoder structure and various attention mechanisms, and achieve new state-of-the-art performance.

Typical approach (BR), unlike seq2seq model, dose not produce cumulative error and the prediction of a single label will not be disturbed by all labels. But on the other hand, this independence causes it to ignore correlations between labels, which might weaken the performance of task.

Seq2seq model, however, suffers from several problems. First, labels are predicted by RNN-based decoder relying on a predefined ordering of labels. Previous studies [1], [2], [30] show that label ordering in dataset greatly affects the model performance. Moreover, setting a strict order to acquire the perfect label order is too costly. Second, seq2seq model is trained with the maximum likelihood estimation (MLE) method and the cross-entropy loss function. Labels with more occurrences are more likely to be predicted, and the correlations among labels cannot be captured by the model without label interaction. Third, at one timestep, the probability of all labels is converted only from the low-dimension hidden state of decoder, which fails to make full use of the text. Moreover, as the timestep increases, the error of label prediction will accumulate when a previous timestep cannot correctly predict a label.

To address these issues, we propose a novel framework that combines the advantages of BR and neural networks. In other words, some relatively independent classifiers are learned by BR, and the fitting ability of neural network is also utilized to extract some global and local information. With MLTC problem been decomposed into binary classification for each label, recurrent neural network (RNN) and convolutional neural networks (CNN) is selected to fully extract the information from the text. As shown in Figure 1, Global and local extractors using neural networks have extracted the information that they care about respectively, which will greatly help to improve the performance of MLTC task. In this method, since no generative method is used to get the label sequence, the relevance of labels is captured through *Extractor* without cumulative error.

In addition, we observe that there are category relationships and hierarchical relationships among labels. Intuitively, in

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

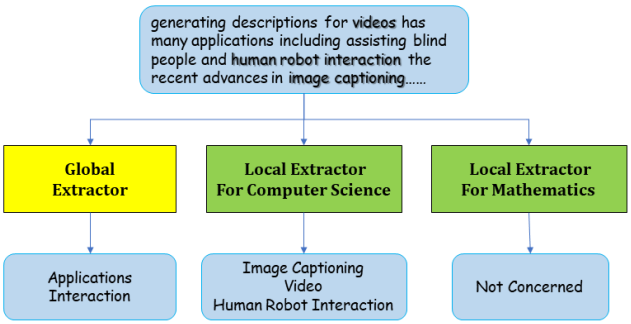

Fig. 1. A simplified example of how global and local extractors operate. *Global Extractor* is utilized to extract basic information useful for the classification task. *Local Extractor for Computer Science* is used to extract interesting information from sentence in current subject. *Local Extractor for Computer Mathematics* also plays the same role. In this example, the first two extractors have found the information they are concerned about, while the text does not involve mathematics-related content, so the third extractor is not concerned.

MLTC task, the quantity of label is large and often artificially pre-specified. There will be an objective hierarchical relationship between labels, or labels can be divided into multiple categories. Labels in the same category share some knowledge. Therefore, we assign a specific local extractor to each category, so that this extractor can accurately extract the information that the category cares about from the sentence. Ideally, as shown in Figure 1, when computer science-related information appears in the text, the corresponding local extractor can discover it.

Inspired by these, we propose the **G**lobal-**L**ocally **En**coding (*GLEN*) Model. *GLEN* utilize *global extractor* to learn knowledge among the text that plays a fundamental role in the results of classification. Then some *local extractors* (category-specific) are used to extract different information of categories.

The experimental results on real-world datasets including AAPD [31], RCV1-V2 [6] show that our model achieves the state-of-art performance compared to all baseline methods. *GLEN* obtains 0.720 micro-F1 and 0.0236 hamming loss on AAPD dataset, outperforming prior state-of-the-art work by 3.0% and 6.0% .On RCV1-V2 dataset, our model achieves 0.870 micro-F1 and 0.0080 hamming loss..

The contributions of this paper are listed as follows:

- Unlike recent efforts to improve the seq2seq model, we propose a new framework which combines the merits of neural network and BRs methods. It captures the relevance of labels and will not produce cumulative error.
- To the best of our knowledge, we are the first to take the category of labels into account in MLTC task. We use global and local extractors to obtain knowledge shared by all categories and knowledge within categories. And the trick of label statement is used to integrate the hierarchical relationship among labels.
- Experimental results demonstrate that our model outperforms all baseline models and achieves the state-of-the-art performance on the dataset AAPD and RCV1-V2.

II. Related Work

Multi-label text classification is one of the most important task in NLP. Many efforts have been invested in it. Early work on exploring the MLTC task focuses on machine learning algorithms, mainly including problem transformation methods and algorithm adaptation methods [2]. Problem transformation methods, such as Binary Relevance (BR) [33], Label Powerset (LP) [34], and Classifier Chains (CC) [35], map the MLTC task into multiple single-label learning task. Algorithm adaptation method extend specific learning algorithm to handle multi-label data directly. The corresponding representative work is ML-DT [27], Rank-SVM [26], ML-KNN [16] and so on.

Typical method -BR is one of the simplest and most popular transformation methods. Its main drawback is that it does not consider dependencies between labels. However, it has has been shown that in many cases, BR can yield predictive performance as good as more complex methods depending on the characteristics of the data [20], [21].

Recent years, the research turned to the application of neural networks. Zhang [36] proposed a neural network algorithm named BP-MLL, which is the multi-label version of backpropagation . Kurata [37] proposed a neural network initialization method to treat some of the neurons in the final hidden layer as dedicated neurons.

Since RNN-based approach (e.g. seq2seq) works well on MLTC task, more and more researchers focus on it. Chen [39] proposed an ensemble application of CNN and RNN to capture both the global and the local textual semantics. The model belongs to encoder-decoder pattern, in which CNN acts as encoder and RNN is used for decoding. Yang [2] and Li [42] proposed to view the multi-label classification task as a sequence generation problem. A seq2seq model with a novel decoder (global embedding) structure is proposed by Yang [2]. Li [42] proposed a label distributed seq2seq model with a novel soft loss function to solve the problem.

To further improve the seq2seq model, some methods of attention mechanism have been proposed. Lin [41] proposed a semantic-unit-based dilated convolution model based on seq2seq. A corresponding hybrid attention mechanism and multi-level dilated convolution are implemented to extract both the information at the word-level and the level of the semantic unit. Wang [44] proposed ranking-based AutoEncoder with a word-vector-based self-attention.

However, some problems inherent in seq2seq are difficult to solve. Training a RNN decoder requires a predefined order of label. Although Nam [29] and Yang [31] compare several ordering strategies and suggest ordering positive labels by frequency directly in descending order (from frequent to rare labels), it is unnatural to impose a strict order on labels, which may break down label correlations in a chain [30]. Yang [2] proposed a novel sequence-to-set framework utilizing deep reinforcement learning, which reduce the dependence on the label order. Tsai [30] proposed a framework based on optimal completion distillation and multitask learning to solve this. Chen [3] proposed the order-free RNN to dynamically decide

a target label at each time during training by choosing the label in the target label set with the highest predicted probability. However, these methods only alleviate some of the problems of the seq2seq model and do not significantly improve the performance of the task.

Therefore, we try to combine the advantages of BR and neural networks to make MLTC tasks achieve better results. In other words, our model is able to achieve relative independence between categories, and can learn the knowledge within the category through the extractors, without being bothered by label ordering and error accumulation.

Besides, we observed that there is a relationship among the labels. The quantity of label is usually large and often artificially pre-specified. Moreover, in order to make the use of labels more efficient, these labels are usually hierarchical. These levels can be expressed as categories. Figure 2 shows the hierarchy of labels in the AAPD dataset. Intuitively, some labels belonging to a certain category should correspond to some common knowledge. For instance, label *cs.ai* and *cs.cv* share some knowledge in the field of computer science.

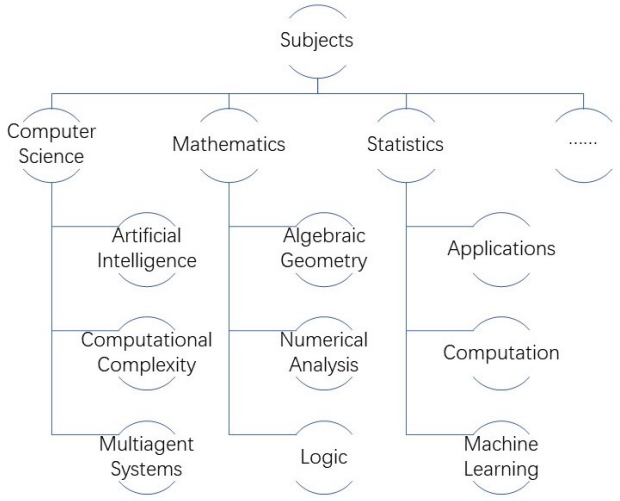

Fig. 2. Hierarchy of labels in AAPD dataset. The actual labels are on the third level of the tree. For example, the label for *Artificial Intelligence* in the field of *Computer Science* is **cs.ai**.

The previous work ignores the fact that these intrinsic information may help for the classification performance. They [30], [37], [41] treat the labels as being at the same level. Inspired by the success of Zhong [28] in dialogue state tracking task, which is a vital component in the task-oriented dialog system [17], [18], [25], we began to pay attention to the use of specific local information in MLTC task.

III. Method

In this section, we will introduce our model in details. First, we give an overview of the model in Section 3.1. Second, we describe how to assign categories to labels in Section 3.2. Third, we explain the details of the *Global Encoding Module* in Section 3.3. Fourth, we introduce the *Local Encoding Module* in Section 3.4. Finally, we present the *Scroing Module* in Section 3.5.

Formally, the source sentence $x = \{x_1, \ldots, x_i, \ldots, x_n\}$ and the target label sequence $t = \{t_1, \ldots, t_i, \ldots, t_m\}$ are both given in the dataset, where n,m are the corresponding lengths. x_i is a word in the sentence, and t_i is a label. Given the label space with L labels, we divide all labels into k categories. For the convenience of explanation, we set k to 4, that is, we assume that there are four categories, and they are recorded as Category A to Category D. The number of elements (labels) in these four categories is l_A, l_B, l_C and l_D. The purpose is to correctly predict the involved label based on the sentence.

A. Overview

Our model is shown in Figure 3. Our model consist of three components: *Global Encoding Module*, *Local Encoding Module* and *Scoring Module*.

Given a sentence x, we first compute the probability of l_A labels under category A, and then calculate the probability of other labels under category B to category D.

Under category A, we use *Global Encoding Module* shared by all categories and *Local Encoding Module* specific to category A to encode the sentence x. As for labels, we first expand l_A labels in category A to label statements, and pass through the *Global Encoding Module* and *Local Encoding Module*, and further simplifed by the self-attention layer. Then, the encoding of labels seperately read the encoding of sentence, and fuse the results to get the score of each label in current category by *Scoring Module*. Next, we select the labels that receive a score above a threshold.

After dealing with all categories, we integrate the labels selected by different categories as the final output.

B. Assign a Category to Each Label

We assign a category to each label. In general, the labels are often pre-specified and each label have practical meaning. And when the quantity of labels is large, in order to improve the work efficiency, these labels will be artificially divided into certain categories.

In the AAPD and RCV1 datasets used in our experiments, these categories of information can be explicitly extracted, which will be described in detail in the *Experimental Settings* in Chapter 4. However, previous studies have ignored these categories of information and treated all labels at the same level.

Then, in our model, we regard categories (e.g. label "category *A*") as first-level labels and actual labels (e.g. label "*a1*") as second-level labels.

To make full use of the first and second level labels, rather than simply using a trainable embedding matrix to encode label, we expand the label into an assignment statement so that more information could be obtained by time sequence encoding.

We use the trick of the label statement here. As shown in Figure 3, label *a_1* is expanded to statement,

$$< s > A = a_1 < /s > \quad (1)$$

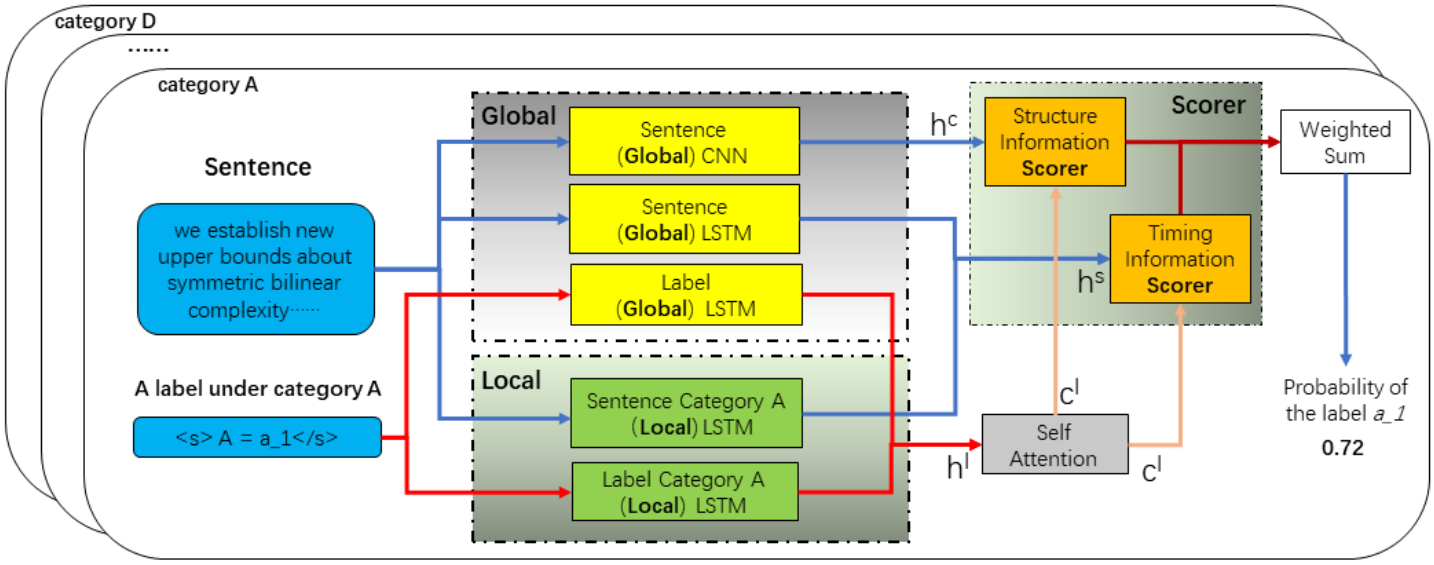

Fig. 3. The overview of our proposed model. Assume that all labels are grouped into 4 categories and they are recorded as Category A to Category D. Our model consists of three components: *Global Encoding Module*, *Local Encoding Module* and *Scoring Module*.

where A is a category with the same dimension as the label embedding. We treat these catgeories as special labels. Specifically, we treat the labels as hierarchical structures, the category labels are on the first level, and the actual labels are on the second level. In this way, our model can learn not only the knowledge within the category, but also the hierarchy of labels through both global and local fashion.

C. Global Encoding Module

Global Encoding Module is shared by all categories. This module consists of three components: a global sentence CNN (convolutional neural network), a global sentence LSTM (bidirectional Long Short-Term Memory) and a global label LSTM. In order to avoid repeated calculation, we only run the global module once for encoding a sentence x when dealing with different categories.

Consider the global sentence LSTM first. It encodes the source sentence from both directions and generates the hidden states for each word,

$$\overrightarrow{h}_i^g = \overrightarrow{\text{LSTM}^{\text{g}}}\left(\overrightarrow{h}_{i-1}, x_i\right), \overleftarrow{h}_i^g = \overleftarrow{\text{LSTM}^{\text{g}}}\left(\overleftarrow{h}_{i+1}, x_i\right) \quad (2)$$

where the superscript g represents global LSTM. Then the hidden state from both directions at each time step are concatenated ($h_i^g = [\overrightarrow{h}_i^g; \overleftarrow{h}_i^g]$). The global label LSTM has the same structure.

As for the global sentence CNN, unlike CNN-RNN model [39] in which CNN is used as an encoder, in our model we apply CNN to extract text features and structure information. The way to use CNN is similar to the one used in [9]. Each word in sentence x will look up the randomly initialized word embedding matrix. The sentence is then a concatenation of word vectors w_i. Thus, a sentence x of length n is represented as:

$$S = w_{1:n} = w_1 \oplus w_2 \oplus \ldots \oplus w_n \quad (3)$$

where $\oplus$ is the vector concatenation operator. So, we can treat the sentence as a "image", and perform convolution on it via linear filters. In natural language, several adjacent words are usually used together to form a phrase. Thus, filters with different window sizes are used to extract this information, which may be helpful for classification. We use $w_{i:i+h-1}$ to represent the sub-sentence. A convolution operation involves a filter W_{CNN} and a bias term.

$$o_i = W_{CNN}^T \cdot S_{i:i+h-1} + b \quad (4)$$

where $i = 1 \ldots n - h + 1$, and $\cdot$ is the dot product between the filter vector and the word vector. This filter is applied repeatedly to each possible window of h words in the sentence to produce an feature map $o = [o_1, o_2, \ldots, o_{n-h+1}]$. Then, we apply a $1 - max - pooling$ to each feature map to induce a fixed-length vector. This process is repeated by different filters with different window sizes.

Next, concatenate the output vector of filters and project it into a lower dimensional vector h^c by a full-connect layer, where superscript c indicates **C**NN, as shown in Figure 3.

D. Local Encoding Module

Local Encoding Module consists of two components: a local sentence LSTM and a local label LSTM. Each category has its own unique LSTM network to extract the information it cares about. The local sentence LSTM and the local label LSTM have the same structure as the global LSTM.

Consider the local sentence LSTM first under category A,

$$\overrightarrow{h}_i^A = \overrightarrow{\text{LSTM}^{\text{A}}}\left(\overrightarrow{h}_{i-1}, x_i\right), \overleftarrow{h}_i^A = \overleftarrow{\text{LSTM}^{\text{A}}}\left(\overleftarrow{h}_{i+1}, x_i\right) \quad (5)$$

where the superscript A represents local sentence LSTM. Then the hidden state from both directions at each time step are concatenated ($h_i^A = [\overrightarrow{h}_i^A; \overleftarrow{h}_i^A]$).

The outputs of the two (global and local) LSTMs for sentence are combined through a category-specific scalar parameter α^A to yield a global-local encoding h^s of sentence x, where superscript s indicates **s**entence and α^A is a trainabel parameter. Now h^s denotes that both universal information for classification and unique information for category A are obtained.

$$h^s = \alpha^A h^g + (1 - \alpha^A) h^A \tag{6}$$

Next ,consider the encoding of labels. Similar to the process of sentence encoding, h^l is calculated by passing the label statement to global label LSTM and local label LSTM, where superscript l means **l**abel. The global label LSTM can process all the labels in different categories and learn the general knowledge applicable to classification contained in the label. And the local label LSTM (for category A) is only used to extract the information of the label under the current category.

Then, different from sentence encoding, we compute a self-attention context c^l over h^l aiming to transform h^l into a vector. Self-attention or intra-attention, is a special case of attention mechanism that only requires a single sequence to compute its representation, which has been applied to many tasks [10]–[13]. It also provides a more flexible way to select, represent and synthesize the information of the inputs [14]. In our case, for ith element in h^l,

$$a_i^l = W_{sa} h_i^l + b \tag{7}$$

$$p^l = \mathrm{softmax}\left(a^l\right) \tag{8}$$

where the subscript of W_{sa} indicates **s**elf-**a**ttention.

The self-attention context c^l is then the sum of each element h_i, weighted by the corresponding normalized self-attention score p_i^l.

$$c^l = \sum_i p_i^l h_i^l \tag{9}$$

E. Scoring Module

Intuitively, we can determine whether the sentence expressed the label under one category by examining two input source, h^c and h^s. The first source is the timing encoding of the sentence x. Labels are interested in words at certain positions in the sentence.

$$a_i^t = \left(h_i^s\right)^\top c^l \tag{10}$$

$$p^t = \mathrm{softmax}\left(a^t\right) \tag{11}$$

$$q^t = \sum_i p_i^t h_i^s \tag{12}$$

$$y^t = W_t q^t + b \tag{13}$$

where t indicates **t**iming information. The score y^t indicates the degree to which the label was expressed by the sentence x under certain category.

The second source is the output of *structure information encoding module* h^c. This source studies the association between label and adjacent words collocations. All labels under the current category c^l get the probability of their classification by reading the structural information h^c of the sentence .

$$y^c = \left(c^l\right)^\top h^c \tag{14}$$

The final score y is then a weighted sum between two source y^t and y^c, normalized by the sigmoid function σ.

$$y = \sigma\left(y^t + u y^c\right) \tag{15}$$

where the weight u is a adjustable parameter. The dimensions of y^t, y^c, and y are all k, which is the number of labels under one category.

Finally, we use binary cross-entropy loss to estimate the loss of prediction. It is a sigmoid activation plus a cross-entropy loss. Unlike softmax loss and cross-entropy loss used in generative model [2], it is independent for each label (class), meaning that the loss computed for every class is not affected by other class. That's why it is used in our model, were the insight of an element belonging to a certain class should not influence the decision for another class. Below is the loss calculation between the score of label y under one category and groudtruth t of sentence x.

$$loss = -\frac{1}{k} \sum_{i=1}^{k} t_i \cdot \log\left(y_i\right) + \left(1 - t_i\right) \cdot \log\left(1 - y_i\right) \tag{16}$$

IV. Experiment

In the following, we evaluate our proposed model on two datasets. We first introduce the dataset, evaluation metrics, all baselines, experimental settings and results. Finally, we perform ablation experiments and discussion.

A. Datasets

Arxiv Academic Paper Dataset (AAPD): This dataset is provided by Yang [31]. It consists of the abstract and corresponding subjects of 55,840 papers in arxiv. An academic paper may have multiple subjects and there are 54 subjects in total, such as *cs.IT*, *math.CO*, *math.IT*, *quant-ph*. To be specific, the training set contains 53840 samples, while the validation set and test set contain 1000 samples respectively. The statistic information of the two datasets is shown in Tabel II.

TABLE I
THE DESCRIPTION OF LABEL OF RCV1-V2 DATASET

Label	Description
C11	Strategy, new companies, joint ventures, consortia, diversifications, investment.
E12	Monetary/economic policy and intervention, interest rates.
G15	All European Community affairs.
M11	Stock exchanges, performance of equities.

Reuters Corpus Volume I (RCV1-V2) [1]**:** This dataset is provided by Lewis [6]. It consists of over 800,000 manually

[1]http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

TABLE II
STATISTIC OF TWO DATASETS. TOTAL SAMPLES, LABEL SETS DENOTE THE TOTAL NUMBER OF SAMPLES AND LABELS. WORDS/SAMPLE IS THE AVERAGE NUMBER OF WORDS PER SAMPLE AND LABLES/SAMPLE IS THE AVERAGE NUMBER OF LABELS PER SAMPLE.

Dataset	Total Samples	Label Sets	Words/Sample	Labels/Sample
RCV1-V2	804,414	103	123.94	3.24
AAPD	55,840	54	163.42	2.41

categorized newswire stories made available by Reuters Ltd for research purposes. Multiple topics can be assigned to each newswire story and there are 103 topics in total. To be specific, the training set contains 802414 samples, while the validation set and test set contain 1000 samples respectively. The topics in the dataset are desensitized, such as *C11*, *C12*, *E521*, *ECAT*. However, the actual meaning of each topic (text interpretation) as shown in Tabel I and the tree structure of the topics are additionally given in the data set. This means that there is a hierarchical relationship between the topics.

B. Evluation Metrics

Following the previous studies [16], [39], we adopt hamming loss and micro-F1 score to evaluate the performance of our models. For reference, the micro-precision as well as micro-recall are also reported.

Hamming-loss [31], [38], [41] evaluates the fraction of misclassified instance-label pairs, where a relevant label is missed or an irrelevant is predicted.

$$HL = \frac{1}{L}\sum \mathbb{I}(y \neq \hat{y}) \tag{17}$$

Micro-F1 [31], [40] can be interpreted as a weighted average of the precision and recall. It is calculated globally by counting the total true positives, false negatives, and false positives.

$$\mathrm{microF}_1 = \frac{\sum_{j=1}^{L} 2tp_j}{\sum_{j=1}^{L} 2tp_j + fp_j + fn_j} \tag{18}$$

C. Baselines

In the following, we introduce the baseline models with which our model compares.

- **Binary Relevance (BR)** [33] transforms the MLC task into multiple single-label classification problems by ignoring the correlations between labels.
- **Classifier Chains (CC)** [35] transforms the MLC task into a chain of binary classification problems and takes high-order label correlations into consideration.
- **Label Powerset (LP)** [34] transforms a multi-label problem to a multi-class problem with one multi-class classifier trained on all unique label combinations.
- **CNN** [9] uses multiple convolution kernels to extract text features, which are then inputted to the linear transformation layer followed by a sigmoid function to output the probability distribution over the label space.
- **CNN-RNN** [39] proposes a CNN and RNN based method that is capable of efficiently representing textual features and modeling highorder label correlation.
- **Seq2Seq** [31] applies the sequence-to-sequence model to perform multi-label text classification.

D. Experimental Settings

We implement our experiments in PyTorch on an NVIDIA 1080Ti GPU. The size of the vocabulary is 50000 and out-of-vocabulary (OOV) words are replaced with unk for both datasets. We use the Adam [8] optimization method to minimize the binary cross-entropy loss over the training data.Follow [31], for the hyper-parameters of the Adam optimizer, we set the learning rate $\alpha = 0.001$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively, and $\epsilon = 1 * 10^{-8}$. We set dropout [7] to 0.2.

For the RCV1-V2 dataset, each document is truncted at the length of 500. We set the (randomly intialized) word embedding size to 256. As for the number of categories, no specific information is displayed in the label, such as *C11*, *E12*, *G15* and *M11*. However, as shown in Table I, each label has an actual text interpretation, and all labels are tree-shaped. We can use this information to assign categories to labels. We use the first letter of label to distinguish categories. Therefore, it can be determined that there are four categories of labels: *C*, *E*, *G*, *M*. All four categories are in the first layer of topic hierachy tree which is shown in official website. For example, topic *ECAT* belongs to category *E*.

Therefore, for the encoding of sentence, one global LSTM and four local LSTM for four categories are used. For the encoding of labels, the same number of LSTMs are configured. The hidden size of the bi-LSTM is 128. It should be noted that the dimension of word embedding must be twice the dimension of hidden size of bi-LSTM due to the inner product of the vector. In *s*tructure information encoding module. we use 100 filters of each the four window-size (4,5,6,7), which result in a 400-dimension output after concatenation. And this output be projected into a 256-dimension vector h^c by a full-connect layer. We set the parameter u in equ.15 to 0.5.

For the AAPD dataset, we use 100 filters of each the four window-size (2,3,4,5). As for the number of categories, we use the prefix of the label (subject) to determine the category. We divided the subjects into three categories: *cs*, *math*, and *the rest*. For example, subject *cs.cv* starts with *cs.*, so it is assigned to category *cs*. Subject *physics.soc-ph* is classified as category *the rest*. To be specific, among the total of 54 subjects, there are 33 subjects in category *cs* and 8 subjects in category *math*, while category *the rest* contains 13 subjects. The size of (randomly initialized) label embedding is 256. Hence, one global LSTM and three local LSTM are configured for the sentence encoding and label encoding. The rest of the parameters are the same as that in the RCV1-V2 dataset.

E. Results

The experiment results of our model and the baselines on both datasets are shown in Tabel IV and Tabel III.

TABLE III
PERFORMANCE ON THE AAPD TEST SET. HL, P, R, AND F1 DENOTE HAMMING LOSS, MICRO-PRECISION, MICRO-RECALL AND MICRO-F1, RESPECTIVELY. THE SYMBOL "+" INDICATES THAT THE HIGHER THE VALUE IS, THE BETTER THE MODEL PERFORMS. THE SYMBOL "-" IS THE OPPOSITE.

Model	HL(-)	P(+)	R(+)	F1(+)
BR	0.0316	0.644	0.648	0.646
CC	0.0306	0.657	0.651	0.654
LP	0.0312	0.662	0.608	0.634
CNN	0.0256	0.849	0.545	0.664
CNN-RNN	0.0278	0.718	0.618	0.664
Seq2Seq	0.0251	0.746	0.659	0.699
Our model	**0.0236**	**0.770**	**0.677**	**0.720**

TABLE IV
PERFORMANCE ON THE RCV1-V2 TEST SET.

Model	HL(-)	P(+)	R(+)	F1(+)
BR	0.0086	0.904	0.816	0.858
CC	0.0087	0.887	0.828	0.857
LP	0.0087	0.896	0.824	0.858
CNN	0.0089	0.922	0.798	0.855
CNN-RNN	0.0085	0.889	0.825	0.856
Seq2Seq	0.0081	0.887	**0.850**	0.869
Our model	**0.0080**	**0.926**	0.834	**0.871**

TABLE V
ABLATION TEST ON AAPD DATASET

Model	HL(-)	P(+)	R(+)	F1(+)
Our model	0.0236	0.770	**0.677**	**0.720**
- *text global-cnn*	**0.0024**	0.781	0.645	0.706
- *text global-local-rnn*	0.0253	**0.803**	0.578	0.672
- *label global-local-rnn*	0.0238	0.779	0.654	0.711

Table III presents the experimental results on AAPD test set. It indicates that our model achieves a reduction of 6.0% hamming loss and an improvement of 3.0% micro-F1 score over the state-of-the-art performance. Similar to the results on the AAPD test set, our model still achieves the prior best work with less parameters on RCV1-V2 dataset.

F. Ablation Test and Discussion

Considering the feasibility of computing resources, we perform some ablation experiments on the AAPD dataset to analyze the effectiveness of different components of our *GLEN* model. The results of these experiments are shown in Tabel V.

In the table, we mark the process of global sentence CNN as *text global-cnn*, the global and local sentence LSTM as *text global-local-rnn* and the process of label statement encoding as *label global-local-rnn*, and . And - *text global-cnn* in the table means that we do not implement *text global-cnn* module in our model, and the rest of our model are working properly.

Labels with categories can improve classification tasks. This is due to the use of categories which get more information within the category mined. In addition, a large classification task is decomposed into some smaller classification tasks under each category, which further reduces error.

Decomposing multi-label classification task into multiple binary classification tasks can effectively avoid the accumulation of errors caused by the generative method (seq2seq). The way to avoid the disadvantage of seq2seq is to apply the traditional classification loss function. Therefore, we use binary cross-entropy loss (BCE) in our model, which leads to an improvement in micro-precision (P) metric. This is due to the fact that some labels with low probability will be trained to tend to 0, the most directly manifestation of which is that the model will not output its own uncertain results. At the beginning of training stage, the prediction is usually empty.

Temporal order is important than structure information. As shown in Tabel V, the performance of the model without *text global-local-rnn* is weaker than the model without *text global-cnn*, which suggest that capturing temporal dependencies is helpful for understanding phrases for classification. Because of the cooperation between two parts, *text global-cnn* and *text global-local-rnn*, our model can achieve the desired results.

Label encoding with label statement trick and self attention works better than ordinary label embedding. We observe that there is a significant decrease in performance when remove the *label global-local-rnn*. This stems from the flexibility in the attention context computation afforded by the self-attention mechanism, which allows the model to focus on selecting sentence history relevant to the current label. Moreover, label statement trick is integrated into the primary and secondary labels, which makes the label encoding process more efficient

V. CONCLUSION

In this paper, we propose a simple and effective novel framework, named **G**lobal-**L**ocally **En**coding (*GLEN*). Experimental results shows taht *GLEN* achieves the state-of-the-art performance. Our model learns information within categories by encoding sentences and labels in both global and local fashion. In addition, through the method of label statement, we integrated the hierarchical relationship of labels into the model for learning. And our model does a good job of avoiding the problems that might arise with a generative approach, such as error accumulation and unclear label ordering.

VI. ACKNOWLEDGEMENT

This paper is supported by the Natural Science Foundation of China (Grant No: U1811462), the Collaborative Innovation Center of Novel Software Technology and Industrialization at Nanjing University.

REFERENCES

[1] Vinyals, O., Bengio, S., & Kudlur, M. (2015). Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391.

[2] Yang, P., Ma, S., Zhang, Y., Lin, J., Su, Q., & Sun, X. (2018). A Deep Reinforced Sequence-to-Set Model for Multi-Label Text Classification. arXiv preprint arXiv:1809.03118.

[3] Chen, S. F., Chen, Y. C., Yeh, C. K., & Wang, Y. C. F. (2018, April). Order-free RNN with visual attention for multi-label classification. In Thirty-Second AAAI Conference on Artificial Intelligence.

[4] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

[5] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[6] Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research, 5(Apr), 361-397.

[7] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[8] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[9] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[11] Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304.

[12] Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933.

[13] Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733.

[14] Tan, Z., Wang, M., Xie, J., Chen, Y., & Shi, X. (2018, April). Deep semantic role labeling with self-attention. In Thirty-Second AAAI Conference on Artificial Intelligence.

[15] Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. In Advances in neural information processing systems (pp. 473-479).

[16] Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40(7), 2038-2048.

[17] Wu, C. S., Socher, R., & Xiong, C. (2019). Global-to-local Memory Pointer Networks for Task-Oriented Dialogue. arXiv preprint arXiv:1901.04713.

[18] Mrkšić, N., Séaghdha, D. O., Wen, T. H., Thomson, B., & Young, S. (2016). Neural belief tracker: Data-driven dialogue state tracking. arXiv preprint arXiv:1606.03777.

[19] Kumar, A., Vembu, S., Menon, A. K., & Elkan, C. (2012, September). Learning and inference in probabilistic classifier chains with beam search. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 665-680). Springer, Berlin, Heidelberg.

[20] Melo, A., & Paulheim, H. (2019). Local and global feature selection for multilabel classification with binary relevance. Artificial intelligence review, 51(1), 33-60.

[21] Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. Pattern recognition, 45(9), 3084-3104.

[22] Zhang, M. L., Li, Y. K., Liu, X. Y., & Geng, X. (2018). Binary relevance for multi-label learning: an overview. Frontiers of Computer Science, 12(2), 191-202.

[23] Senge, R., del Coz, J. J., & Hüllermeier, E. (2019). Rectifying classifier chains for multi-label classification. arXiv preprint arXiv:1906.02915.

[24] Li, N., & Zhou, Z. H. (2013, May). Selective ensemble of classifier chains. In International Workshop on Multiple Classifier Systems (pp. 146-156). Springer, Berlin, Heidelberg.

[25] Sharma, S., Choubey, P. K., & Huang, R. (2019). Improving Dialogue State Tracking by Discerning the Relevant Context. arXiv preprint arXiv:1904.02800.

[26] Elisseeff, A., & Weston, J. (2002). A kernel method for multi-labelled classification. In Advances in neural information processing systems (pp. 681-687).

[27] Clare, A., & King, R. D. (2001, September). Knowledge discovery in multi-label phenotype data. In European Conference on Principles of Data Mining and Knowledge Discovery (pp. 42-53). Springer, Berlin, Heidelberg.

[28] Zhong, V., Xiong, C., & Socher, R. (2018). Global-locally self-attentive dialogue state tracker. arXiv preprint arXiv:1805.09655.

[29] Nam, J., Mencía, E. L., Kim, H. J., & Fürnkranz, J. (2017). Maximizing subset accuracy with recurrent neural networks in multi-label classification. In Advances in neural information processing systems (pp. 5413-5423).

[30] Tsai, C. P., & Lee, H. Y. (2019). Order-free Learning Alleviating Exposure Bias in Multi-label Classification. arXiv preprint arXiv:1909.03434.

[31] Yang, P., Sun, X., Li, W., Ma, S., Wu, W., & Wang, H. (2018). Sgm: sequence generation model for multi-label classification. arXiv preprint arXiv:1806.04822.

[32] Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research, 5(Apr), 361-397.

[33] Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. Pattern recognition, 37(9), 1757-1771.

[34] Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM), 3(3), 1-13.

[35] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. Machine learning, 85(3), 333.

[36] Zhang, M. L., & Zhou, Z. H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. IEEE transactions on Knowledge and Data Engineering, 18(10), 1338-1351.

[37] Kurata, G., Xiang, B., & Zhou, B. (2016, June). Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 521-526).

[38] Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. Machine learning, 37(3), 297-336.

[39] Chen, G., Ye, D., Xing, Z., Chen, J., & Cambria, E. (2017, May). Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2377-2383). IEEE.

[40] Schütze, H., Manning, C. D., & Raghavan, P. (2008, June). Introduction to information retrieval. In Proceedings of the international communication of association for computing machinery conference (Vol. 4).

[41] Lin, J., Su, Q., Yang, P., Ma, S., & Sun, X. (2018). Semantic-unit-based dilated convolution for multi-label text classification. arXiv preprint arXiv:1808.08561.

[42] Li, W., Ren, X., Dai, D., Wu, Y., Wang, H., & Sun, X. (2018). Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions. arXiv preprint arXiv:1808.05437.

[43] Wang, B., Chen, L., Sun, W., Qin, K., Li, K., & Zhou, H. (2019). Ranking-Based Autoencoder for Extreme Multi-label Classification. arXiv preprint arXiv:1904.05937.

[44] Tsai, C. P., & Lee, H. Y. (2019). Order-free Learning Alleviating Exposure Bias in Multi-label Classification. arXiv preprint arXiv:1909.03434.