IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2019, accepted October 18, 2019, date of publication October 22, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948855

Hierarchical Sequence-to-Sequence Model for Multi-Label Text Classification

ZHENYU YANG AND GUOJING LIU
School of Computer Science and Technology, Qilu University of Technology (ShanDong Academy of Sciences), Jinan 250353, China
Corresponding author: Zhenyu Yang (yang_zhenyu@163.com)

This work was supported in part by the Shandong Natural Science Fund Project under Grant ZR2017LF021.

ABSTRACT We propose a novel sequence-to-sequence model for multi-label text classification, based on a ''parallel encoding, serial decoding'' strategy. The model combines a convolutional neural network and self-attention in parallel as the encoder to extract fine-grained local neighborhood information and global interaction information from the source text. We design a hierarchical decoder to decode and generate the label sequence. Our method not only gives full consideration to the interpretable fine-gained information in the source text but also effectively utilizes the information to generate the label sequence. We conducted a large number of comparative experiments on three datasets. The results show that the proposed model has significant advantages over the state-of-the-art baseline model. In addition, our analysis demonstrates that our model is competitive with the RNN-based Seq2Seq models and that it is more robust at handling datasets with a high label/sample ratio.

INDEX TERMS Sequence-to-sequence, multi-label text classification, self-attention, hierarchical decoder, attention mechanism.

I. INTRODUCTION

Multi-label text classification [1], [2] is an important and challenging task in natural language processing (NLP), that is more complicated than single-label classification because labels often exhibit complex dependencies. A real life, typical example is that terms such as ''politics'', ''economics'', ''culture'' and other labels often appear on the front pages of news websites. The goal is to aid users in selecting the information they desire without being presented with irrelevant information.

As a significant NLP task, many methods have been proposed and have gradually achieved satisfactory performances. Binary relevance (BR) [3] is one of the earliest methods; it models the task as consisting of multiple single-label classification problems by actively ignoring the label dependencies to achieve a certain level of performance. To capture the label dependencies, a classifier chain (CC) [4] is used to convert the task into a series of binary classification problems and model the dependencies. Conditional random fields (CRF) [5] and conditional Bernoulli mixtures (CBM) [6] have also been utilized to handle label dependencies. However, the above methods are applicable only for small or medium-scale datasets, which makes them difficult to apply to large-scale datasets.

With the development of neural networks, some neural models have been applied to solve this task that have achieved improvements. The model proposed in [7] utilizes word embedding and a convolutional neural network (CNN) to capture the label dependencies and address this task, while the model proposed in [8] extracts global and local semantic information from the text through a CNN and a recurrent neural network (RNN). The authors of [9] proposed a deep neural network (DNN)-based model, called a Canonical Correlated AutoEncoder (C2AE). However, these methods insufficiently consider the problem of capturing label dependencies and extracting interpretable information to perform classification from source text.

To better solve the multi-label text classification problem, [10] skillfully uses the sequence-to-sequence (Seq2Seq) model, which shines on neural machine translation (NMT) tasks [11]–[14]. As applications involving long short-term memory (LSTM) networks became more widespread [15], the LSTM-based Seq2Seq [16] model with an attention mechanism was proposed to further improve the performance on this task. For multi-label text classification, Seq2Seq can encode a given source text and decode the representation to form a new sequence that approximates the label sequence. Using the attention mechanism, the decoder effectively

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

extracts important source text information, thereby improving the decoding quality. The decoder utilizes the RNN to generate labels sequentially and to predict the next label based on the previously predicted labels. Therefore, the dependencies between labels must be modeled accurately. However, [17] proved that the attention mechanism does not play a significant role in this task. We perform a deep analysis and find that the word-level information extracted by an LSTM is not well suited for the multi-label text classification task, which leads to inefficiency in the attention mechanism, causing it to have difficulty. In Subsection VI-A, we validate this idea using an attention experiment.

Therefore, it is especially important to obtain the most useful information from the source text for classification. For multi-label text classification, the usual labeling approach of humans can be divided into two steps: First, a person strives to understand the overall meaning of the text; then, they distribute different labels based on the meanings of the different parts of the text. For example, consider the sentence ''Global sea level rise is due to a large amount of greenhouse gas emissions, and the United States announced its withdrawal from the Paris Agreement, which may further aggravate this problem.''. We can easily extract some word-level information from a global perspective, such as ''greenhouse'', ''emissions'', ''withdrawal'', ''agreement'', and ''aggravate''. Such words often play an important role in the overall understanding of the text, so we term such words as :''global interactive information''. From a local perspective, the information in the text can be summarized into the following two aspects: ''emission of greenhouse gases'' and ''US announced its withdrawal from the Paris Agreement''; we call these ''local neighborhood information''. These two parts determine that the text can be classified into two categories, ''environment'' and ''politics''.

From the above analysis, the local neighborhood information is the key to classifying text and assigning appropriate text labels. The local neighborhood information functions as a high-level summary of the local context in text, and it is obviously more suitable for classification. In contrast, global interaction information solely plays the role of aiding overall understanding.

In this paper, we continue the basic form of the Seq2Seq model in our work by considering a multi-label text classification task from the perspective of label sequence generation. We summarize and propose a sequence-to-sequence learning strategy called ''parallel encoding, serial decoding'' and design a novel Seq2Seq model. The model consists of an encoder and a decoder with an attention mechanism. Instead of using LSTM that is not accurately modeled, the encoder uses a CNN and self-attention to obtain fine-grained information from both local and global perspectives. A CNN [18], [19] can efficiently obtain local neighborhood information in parallel, such as from phrases or sentences. Self-attention [20] is an advanced technology that captures global interactions in parallel and ameliorates the global dependency problem. For the decoder, to efficiently utilize the two types information extracted by the encoder described above, we design a hierarchical decoding structure that consists of two decoding blocks connected in series, using an LSTM as the base cyclic unit. The attention mechanism is applied to the respective decoding block and can effectively extract important information, thereby improving the quality of the entire decoder output.

In brief, our contributions are described below:

- We analyze the importance of interpretable local and global information in source text for multi-label text classification and propose a strategy called ''parallel encoding, serial decoding''.
- We design a novel Seq2Seq model that not only accurately extracts the interpretable local and global information used for classification, but also effectively utilizes both types of information while maintaining the independence and integrity of each type.
- The experimental results show that the performance of the proposed model is better than that of the state-of-the-art methods. The proposed model achieves superior results on several large-scale datasets. In addition, the analysis demonstrates that our model with fine-grained information extracted from source text performs better on datasets with high label/sample ratio.

The whole paper is organized as follows. We introduce related work in Section II, and we describe our method in Section III. In Section IV, we design the experiment. In Section V and VI, we present a series of experiments and make analysis and discussions. In final Section VII, we conclude this paper and explore the future work.

II. RELATED WORK

The current models for solving multi-label text classification tasks can be classified into three main categories: problem transformation methods, algorithm adaptation methods and neural network models.

Problem transformation is the simplest method: it converts a multi-label text classification task into multiple single-label learning tasks. BR [3] directly chooses to ignore the label dependencies and build a separate classifier for each label. To capture the label dependencies, label powerset (LP) [2] turns this task into a multi-classification problem for label combinations by using a unique binary classifier for each label combination. CC [4] converts the task into a chain of binary classification problems in which subsequent binary classifiers are based on previous predictions.

Algorithm adaptive methods address multi-label text classification tasks by modifying specific algorithms. For example, ML-DT [21] performs classification by constructing a decision tree based on multi-label entropy; Rank-SVM [22] adopts a support vector machine (SVM) similar to a learning system to handle multi-label problems; [23] proposed an ML-KNN model to determine the label set for each sample using the k-nearest neighbor algorithm and maximal posterior probability; [24] sorts the collection of labels by comparing label pairs; [5] and [25] apply CRF for this task;

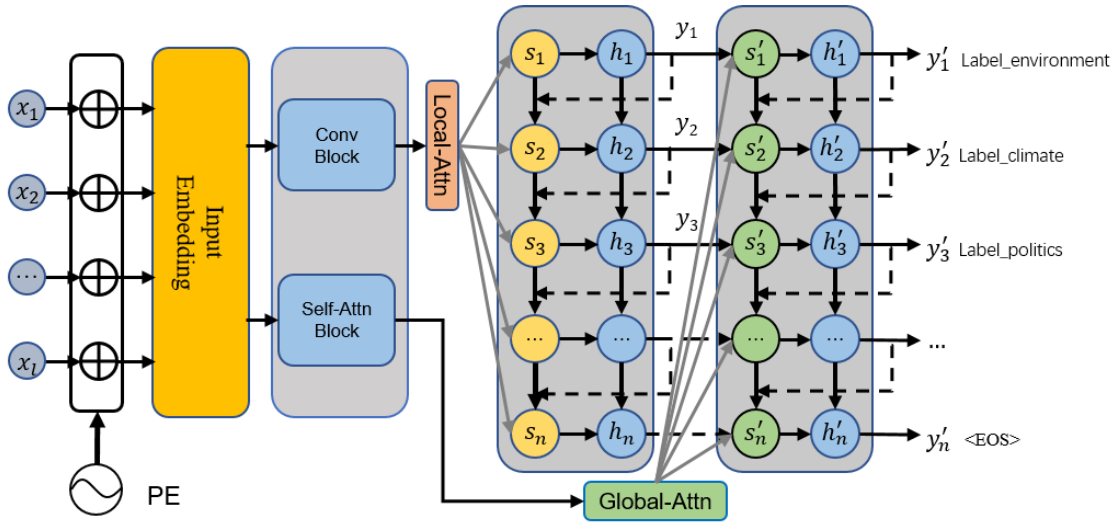


FIGURE 1. The overview of our proposed model. PE denotes the position embedding. Conv Block denotes the convolution block. Self-Attn Block denotes the self-attention block. Local-Attn denotes the local neighborhood information attention. Global-Attn denotes the global interactive information attention.

and [6] uses a CBM to simplify the task by transforming it into multiple standard binary and multiclass problems for classification prediction.

In recent years, neural networks have achieved great success in the NLP field. Specifically, for multi-label text classification tasks, [26] proposed the BP-MLL algorithm, which utilizes a fully connected neural network and a pairwise ranking loss function; [27] applies a better cross-entropy loss function instead of a pairwise ranking loss function; [8] combined a CNN and an RNN to capture local and global semantic information and model high-order correlations between labels. The SGM [16] and MDC [17] both use the LSTM-based Seq2Seq structure: one applies a novel decoder with global embedding and the other includes an additional semantic unit with hybrid attention to create an information-enhanced representation.

III. METHOD

In this section, we introduce our method in detail. First, we provide an overview of the model in Subsection III-A. Then, we explain the details of the encoder in Subsection III-B. Finally, Subsection III-C presents our novel hierarchical decoder with attention.

A. OVERVIEW

We first define some notations and describe the multi-label text classification task. Given a label space Y with N labels $Y = \{y_1, y_2, \ldots, y_N\}$ and a text sequence x, the task is to assign a set of labels y (*quantity* ≥ 2) to the text sequence x. The label sequence generation task can be specifically modeled to find an optimal label sequence y^* that maximizes the conditional probability $p(y|x)$, which is calculated as follows:

$$p(y|x) = \prod_{i=1}^{n} p(y_i|y_{i-1}, y_{i-2}, \ldots, y_1, x) \tag{1}$$

Let $(w_1, w_2, \ldots, w_L)$ be a text sequence with L words, where w_i is the one-hot representation of the i-th word. We obtain the word representation matrix $e = (e_1, e_2, \ldots, e_L)$, $e_i \in R^d$, for this text sequence from the embedding matrix $E \in R^{d \times |V|}$. Here $|V|$ represents the size of the vocabulary, and d is the dimension of the embedding vector.

Figure 1 shows an overview of our proposed model. First, the convolution block and the self-attention block are used in the encoder to obtain local neighborhood information h^L and global interaction information h^G from the text sequence, respectively. Before x is input into the encoder, we embed the position vector [20], [28] $p = (p_1, p_2, \ldots, p_L)$, $p_i \in R^d$ to make use of the positional relationship of words. The final representation of the text sequence input to the encoder is $x = (e_1 + p_1, e_2 + p_2, \ldots, e_L + p_L)$. Then, the hierarchical decoder performs two consecutive decodings. The first decoding takes the context vector c_t weighted by the local neighborhood information h^L; its hidden state s_{t-1} and the embedding vector $g(y_{t-1})$ at time-step $t-1$ are the inputs used to produce the hidden state s_t of the first decoding at time-step t. The second decoding takes the context vector c'_t weighted by the global interaction information h^G; the hidden state s'_{t-1} and the embedding vector $g'(y'_{t-1})$ of the second decoding at time-step $t-1$ are the inputs used to produce the hidden state s'_t of the second decoding at time-step t. Here, y_{t-1} and y'_{t-1} are the predicted probability distributions over the label space Y at time-step $t-1$. The functions g and g' respectively take y_{t-1} and y'_{t-1} as input to produce the embedding vector, which is then passed to the decoding block. Finally, the masked softmax layer is used to output the probability distribution y'_t.

B. ENCODER

A CNN utilizes a fixed-size input and produces a fixed-size output, which means that the convolution block uses a

fixed-size convolution kernel to scan the entire text sequence to obtain local neighborhood information. The self-attention mechanism considers the interactions between words from a global perspective and it is good at capturing correlations within text sequences. In addition, self-attention can solve the global dependency problem in a parallel fashion. Both the convolution and self-attention blocks share the same embedding vector of the source text as input. Finally, the encoder outputs the local neighborhood information representationh^L and the global interaction information representation h^G in parallel.

1) CONVOLUTION BLOCK

according to previous work of CNN in NLP [18], we use a one-dimensional convolution [19] followed by a nonlinear activation function. The Nonlinear activation function allows the convolution block to fully utilize the input sequence by focusing on less word information if needed and filtering out redundant and extraneous information.

The convolution kernel has a width of k and uses the input sequence $X \in R^{k\times d}$, which consists of a sequence of k consecutive words (such as phrases or sentences). To obtain the high-level local neighborhood information, we use the stacked network and add the residual connection [29] to the block output. We choose a gated linear unit (GLU) [30] as the nonlinear activation function, which implements a simple gating mechanism over the output of the convolution:

$$v([A|B]) = A \otimes \sigma(B) \tag{2}$$

where $A, B, v([A|B]) \in R^d$. $[A|B] \in R^{2d}$ is the output of the one-dimensional convolution, and $\otimes$ is a pointwise multiplication. The $\sigma(B)$ control, whose input is A in the current context, is also relevant.

In addition, to ensure the balanced distribution of input data, we use layer normalization [31] to control the input distribution and variance. The output of the l-th convolution block is calculated as follows:

$$h_i^l = v(W^l h_i^{l-1}) + h_i^{l-1} \tag{3}$$

$$h^L = (h_1^l, h_2^l, \ldots, h_L^l) \tag{4}$$

where $W^l \in R^{2d\times kd}$ is a weight parameter. For simplicity, we omit all the bias terms in this paper. Here, h_i^l is the local neighborhood information around the i-th word.

As the depth of the network increases, the convolution blocks acquire information that is further away from the central word; however, what we need is a high-level representation around the i-th word. For this, we set the convolution kernel width in all the convolution blocks to a monotonically decreasing trend. For example, a convolution block with three layers, the convolution kernel widths are set to [5,3,3]. This approach not only reduces the interference from long-distance context information and enhances the information representation of the local context but also reduces the number of parameters and the computational complexity.

2) SELF-ATTENTION BLOCK

We chose the multi-head self-attention (MHA) [20], which expands the ability of the model to focus on multiple locations within a sequence and refines the representations of words in the global interaction information.

We set the number of heads to M. Here, d_v, d_k and d_q refer to the depths of the values, keys and queries, respectively. We further denote the depths of values, keys, and queries on the m-th head to d_v^m, d_k^m and d_q^m. For a given input sequence $X \in R^{L\times d}$, the MHA output is calculated as follows:

$$head_m = softmax(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}})(XW_v) \tag{5}$$

$$\text{MHA(X)} = Concat[head_1, head_2, \ldots, head_M]W_O \tag{6}$$

where the projections are parameter matrices $W_q \in R^{d\times d_q^h}$, $W_k \in R^{d\times d_k^h}$, $W_v \in R^{d\times d_v^h}$, $W_O \in R^{md_v^m\times d}$.

Similar to a convolution block, we use a stacked network to obtain high-level global interaction information and add residual connection and layer normalization to the network. The output of the g-th self-attention block is calculated as follows:

$$h^G = \text{MHA}(h^{g-1}) + h^{g-1} \tag{7}$$

C. HIERARCHICAL DECODER WITH ATTENTION

To ensure good use of the two kinds of fine-grained information extracted by the encoder, we design a hierarchical decoder, which consists of two decoding blocks in series and utilizes LSTM as the basic cyclic unit. Using the simple and effective ''additive attention'' [11], [13] mechanism, the weights of the two types of information are calculated in their respective decoding blocks.

In the first decoding block, we simply continue the decoding structure of [16]. The ''additive attention'' block takes the local neighborhood information h_i^L of the i-th word, the hidden state s_{t-1} and the embedding vector $g(y_{t-1})$ at time-step $t-1$ as inputs, and weights the label to be predicted at time-step t. Finally, the output of the first decoding block at time-step t is y_t.

In the second decoding block, we improve the ''additive attention''. In addition to the inputs (which are similar to the first decoding block) we also add an additional input y_t, which is the output of the first decoding block at time-step t. The second decoding block assigns the weight α_{ti} to the i-th word at time-step t as follows:

$$e_{ti} = V_a^T \tanh(W_a s_t' + Z_a y_t + U_a h_i^G) \tag{8}$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{i=1}^{L}\exp(e_{ti})} \tag{9}$$

where W_a, Z_a, U_a and V_a are weight parameters. s_t' is the hidden state of the second decoding block. h_i^G is the global interaction information of the i-th word.

The final context vector c'_t, which is passed to the second decoding block at time-step t, is calculated as follows:

$$c'_t = \sum_{i=1}^{L} \alpha_{ti} h_i^G \tag{10}$$

The hidden state s'_t of the second decoding block at time-step t is computed as follows:

$$s'_t = \text{LSTM}(s'_{t-1}, [g'(y'_{t-1}); y_t; c'_{t-1}]) \tag{11}$$

where $[g'(y'_{t-1}; y_t; c'_{t-1})]$ denotes the concatenation of the vectors $g'(y'_{t-1})$, y_t and c'_{t-1}. Note that $g'(y'_{t-1})$ is the embedding of the label that has the highest probability under the distribution y'_{t-1}, while y'_{t-1} is the probability distribution over the label space Y at time step $t-1$, which is computed as follows:

$$o'_t = W_o f(W_d s'_t + V_d c'_t) \tag{12}$$

$$y'_t = softmax(o'_t + I_t) \tag{13}$$

where W_o, W_d and V_d are weight parameters, $I_t \in R^N$ is the mask vector used to prevent the second decoding block from predicting repeated labels, and f is a nonlinear activation function.

$$(I_t)_n = \begin{cases} -\infty & \text{previously predicted labels} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Above, we provided an overview of the model and its internal technical details. In addition, for the label sequence, we use the method of [8] to reorder the label data in descending order of frequency and add *bos* and *eos* symbols to the head and tail of the label sequence, respectively.

During training, the loss function is cross entropy loss, and we also use the beam search algorithm [32] to find the optimal prediction sequence for inference. The prediction paths ending with *eos* are added to the candidate path set.

IV. EXPERIMENTAL DESIGN

We present the datasets and preprocessing used in the experiments in Subsection IV-A. The evaluation metrics for the multi-label text classification task are given in Subsection IV-B. In Subsection IV-C, we report on the appropriate model parameters for the three datasets, and Subsection IV-D shows a set of baseline models for comparison.

A. DATASETS AND PREPROCESSING

1) REUTERS CORPUS VOLUME I (RCV1-V2)[1]

The dataset [14] consists of more than 800 K newswire story stories manually compiled by Reuters Ltd. for scientific research purposes. Most reports include two or more topics; there are 103 topics in total. The training set contains 802,414 samples, and the development and test sets each have 1,000 samples. We filtered out samples with more than 500 words and some anomalies; these operations removed approximately 0.5% of the original sample data from the training set. The vocabulary was set to 50 K words, and we replaced some low-frequency words, abnormal words and out-of-vocabulary words with "</>" and "<unk>" symbols.

2) ARXIV ACADEMIC PAPER DATASET(AAPD)[2]

The dataset was collected by the authors of [16], and contains summaries of 55,840 papers in the computer science field and related subjects. An academic paper can have multiple subjects, and there are a total of 54 subjects. We also excluded samples with more than 500 words and some anomalies, which removed approximately 0.2% of the samples from the training set. The development set and test set each have 1,000 samples, and the vocabulary size was set to 30 K. Some low-frequency words, abnormal words and out-of-vocabulary words were handled in the same way as with the RCV1-V2 dataset.

3) ZHIHU QUESTION-TOPIC DATASET (ZHIHU-QT)[3]

This dataset is the official dataset used in the "Zhihu Machine Learning Challenge 2017". We collected 101,251 samples containing 119 topics; the average length of each sample was 140 words. We randomly selected two sets of 1,000 samples to form development and test sets. The pretrained vocabulary was set to 36,000 words. Words with frequencies below 5 in the vocabulary were deleted and replaced with "<unk>" in the samples.

B. EVALUATION METRICS

Following the previous work [8], [23], we adopt hamming loss and micro-F_1 score as our main evaluation metrics. For reference, micro-precision and micro-recall are also recorded for further analysis.

- **Hamming loss** [33] evaluates the fraction of misclassified instance-label pairs, where a relevant label is missed or an irrelevant is predicted, which is calculated as follows:

$$HL(y, \hat{y}) = \frac{1}{L} \sum_{j=1}^{L} 1(y_j \neq \hat{y}_j) \tag{15}$$

  where y_j is the true value corresponding to the j-th label, $\hat{y}_j$is the predicted value of the j-th label, L is the number of labels, and *1(x)* is the indication function.
- **Micro-F_1** [34] can be interpreted as a weighted average of the precision and recall. It is calculated globally by counting the total true positives tp_j, false negatives fn_j, and false positives fp_j, which is calculated as follows:

$$\text{Micro} - F_1 = \frac{\sum_{j=1}^{L} 2tp_j}{\sum_{j=1}^{L} \left(2tp_j + fp_j + fn_j\right)} \tag{16}$$

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

[2] https://arxiv.org/

[3] https://biendata.com/competition/zhihu/

C. EXPERIMENT SETTINGS

We implemented our experiments using PyTorch on the NVIDIA 1080Ti GPU.

For the RCV1-V2, AAPD and Zhihu-QT datasets, the embedding size and the number of units in the hidden layers were both 512. The batch size was set to 64. The convolution blocks in the encoder were set to 5, 3 and 3, and the corresponding convolution kernel widths are [7, 5, 5, 3, 3], [5, 3, 3] and [5, 3, 3], respectively. The number of self-attention block in the encoder was set to 9, 6, and 9, and the beam size was set to 9.

We used the Adam optimizer [35] and set the two momentum parameters to $\beta_1 = 0.89$ and $\beta_2 = 0.899$. The initial learning rate was set to 0.0002 based on the model's performance. In addition, we applied dropout regularization [36] to avoid overfitting and gradient clipping [37] with the range $[-10, 10]$.

D. BASELINES

We use the following baselines to compare our models on the RCV1-V2 and AAPD dataset.

- **Binary Relevance (BR)** [3] converts the multi-label text classification task into multiple single-label classification problems.
- **Classifier Chains (CC)** [4] converts the task into a chain of binary classification problems, and uses this to model the dependencies between labels.
- **Label Powerset (LP)** [2] turns the task into a multi-class problem with the label combination, which sets a unique multi-classifier for each label combination.
- **CNN** [18] extracts text features using a deep convolutional network, then inputs them into a linear transformation layer, and then uses the sigmoid function to output a probability distribution over the label space.
- **CNN-RNN** [8] utilizes CNN and RNN to capture both global and local textual semantics information and models the dependencies between labels.
- **S2S + Attn** [12], [13] is implementation of the RNN-based Seq2Seq model with the attention mechanism.
- **SGM** [16] is a label sequence generation model. It is implementation of the LSTM-based Seq2Seq with the attention mechanism. It uses a decoder with global embedding to capture the dependencies between labels.
- **MDC** [17] uses an additional multi-level expansion convolution component to extract high-level semantic information based on the LSTM-based Seq2Seq and uses the corresponding hybrid attention.

Following the previous work [8], we implement BR, CC and LP based on linear SVM classifier on Scikit-Multilearn [38], which is a open-source library for the task. In addition to SGM and MDC using their own settings, we have followed the hyper-parameter settings for CNN, CNN-RNN and S2S + Attn in MDC [17].

TABLE 1. Performance on the RCV1-V2 test set. HL, P, R and F1 denote Hamming loss, micro-precision, micro-recall, and micro-F_1, respectively.

Models	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
S2S+Attn	0.0081	0.889	0.848	0.868
SGM	0.0075	0.897	0.860	0.878
MDC	0.0072	0.891	0.873	0.882
Our Model	**0.0070**	0.899	**0.888**	**0.893**

TABLE 2. Performance on the AAPD test set. HL, P, R and F1 denote hamming loss, micro-precision, micro-recall, and micro-F_1, respectively.

Models	HL(-)	P(+)	R(+)	F1(+)
BR	0.0316	0.664	0.648	0.646
CC	0.306	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
S2S+Attn	0.0261	0.720	0.639	0.677
SGM	0.0245	0.748	0.675	0.710
MDC	0.0240	0.752	0.681	0.715
Our Model	**0.0236**	0.764	**0.690**	**0.725**

V. RESULTS

In this section, we report the comparison of our model and baselines on RCV1-V2 and AAPD.

We present the results of our model and all the baselines on the RCV1-V2 dataset in Table 1. From these experimental results, it can be found that classical models (such as LP and CNN) that are based on machine learning and deep learning, respectively, still have certain advantages even without using the Seq2Seq structure. The Seq2Seq model achieves some improvements in the dataset compared with the above classical models. For instance, the SGM model achieves a reduction in the Hamming loss of 15.7% and an improvement of 2.7% on the micro-F_1 score over the famous CNN model. Our model achieved a more significant advantage based on the primary evaluation metrics. Reducing the Hamming loss by 21.3% and improving the micro-F_1 score by 4.4% over the CNN model. Compared to the state-of-the-art baseline MDC, our model still garners the leading position with regard to the evaluation metrics. These results occur because our model considers both more fine-gained local neighborhood information and global interaction information, and it leverages them in a near-lossless manner. Compared to the baseline MDC, our model reduces the Hamming loss by 2.8% and increases the micro-F_1 score by 1.2%.

We also present the result of comparisons on the AAPD dataset in Table 2. Similar to the results of RCV1-V2, the classic methods (except for the Seq2Seq models) have a certain

level of competitiveness in the main evaluation metrics. In particular, the traditional CNN is ahead of all current baseline models (including ours) on the micro-precision evaluation. In response to this, we plan to conduct further experiments and analysis in future research. Moreover, we found that the primary S2S + Attn model achieves results close to those of the baseline models above. Comparing the details between RCV1-V2 and AAPD, we observe that the number of samples (approximately 16:1) and the number of labels (approximately 2:1) have an impact on model performance. We delved into the subtle relationship between the above parameters and found that the label/sample ratio has a large influence on the multi-label text classification task. We conducted a series of experiments regarding this ratio and analyze and discuss it in Subsection VI-C.

Compared with all the above models, our model considers more of the fine-grained local neighborhood information and global interaction information in the source text. Using hierarchical decoding not only maintains the integrity of the weighting process but also maintains the independence of each type of information. Our model achieves a Hamming loss reduction of 1.7% and a micro-F_1 score improvement of 1.4%, compared with the current advanced baseline MDC. In addition, our proposed model achieves the best performance on the main evaluation metrics, reducing the Hamming loss by 1.7% and improving the micro-F_1 score by 1.4% compared to the current advanced baseline MDC model.

VI. ANALYSIS AND DISCUSSION

We performed some further exploration and analysis of our model on the Zhihu-QT dataset. In Subsection VI-A, we explore the impact of the attention mechanism on our model. In Subsection VI-B, we perform a set of ablation tests on our model, and in Subsection VI-C, we analyze and discuss the label/sample ratio.

A. EXPLORATION OF ATTENTION

To explore the impact of the attention mechanism for our model, we conducted multiple types of attention experiments and performed an experiment both with and without the attention mechanism. We choose the popular ''dot-product attention'' and ''additive attention'' mechanisms and also introduce the ''hybrid attention'' from the MDC [17] for comparison. To perform a fair evaluation, for ''dot-product attention'' and ''additive attention'', we modified the MDC to use our hierarchical decoder for these two decoding types. For ''hybrid attention'', we modified our model to use the original decoder type from MDC, which involves only one decoding operation.

Table 3 shows a comparison of the two models under the various attention mechanisms. The MDC without attention compared to the MDC with the common dot-product attention or additive attention do not differ substantially on the evaluation metrics. However, evaluation metrics for the same comparison shows much greater differences using our model. For instance, additive attention achieves a Hamming loss reduction of 11.4% and a micro-F_1 score improvement of 7.7% over our model without attention. From our analysis, we believe that an LSTM is good at extracting long-distance word-level semantic information from sequences rather than at summarizing fine-grained local information; therefore, it is not suitable for multi-label text classification tasks. Our method involves modeling sequences by mimicking human labeling habits, an approach that can extract interpretable fine-grained information from a sequence. Obviously, such information is more suitable for label prediction, and it improves the efficiency of the attention mechanism. Therefore, our method garners a substantial lead on the evaluation metrics.

TABLE 3. Performance the comparison with different attention mechanisms between our model and MDC. ''w/o attention'' means without using attention, and ''dot attention'' denotes ''dot-product attention''.

Models	Attentions	HL(-)	F1(+)
MDC	+w/o attention	0.0102	0.801
	+dot attention	0.0101	0.804
	+additive attention	0.0103	0.803
	+hybrid attention	**0.0097**	**0.833**
Our model	+w/o attention	0.0105	0.790
	+dot attention	0.0094	0.849
	+additive attention	**0.0093**	0.851
	+hybrid attention	0.0094	**0.852**

A comparison of additive and hybrid attention in the MDC shows that the hybrid attention mechanism achieves a Hamming loss reduction of 4.0% and a micro-F_1 score improvement of 2.7%. This results confirms that the advanced hybrid attention method improves the performance of the MDC. However, the evaluation metrics are not much different in our model. Our study found that MDC focuses on extracting semantic word-level information and improves the utilization of the information by hybrid attention. In contrast, our model considers two more fine-gained information types in the source text that tend to better represent the labeled meaning of the source text and have greater impacts on label sequence prediction. From a subtler perspective, additive attention functions slightly better than does hybrid attention in our model. This result not only shows that the ordinary attention mechanism remains efficient in our model but also indicates that a powerful information capture capability is the key to improving the performance of a multi-label classification model.

B. ABLATION TEST

To fully evaluate the effects of our proposed model, we performed a set of ablation tests with our model. We controlled the variables by removing some of the modules from the model to enable a comparison of their effects. Specifically, in addition to the additive attention evaluation discussed in Subsection VI-A, we also conducted three sets of experiments to evaluate the contributions of the convolution block, the self-attention block and the hierarchical decoder to the

TABLE 4. Performance of each module on the Zhihu-QT test set. "C-Block" denotes the model without the convolution block. "SA-Block" denotes the model without the self-Attention block. "H-Decoder" denotes the model without hierarchical decoder.

Models	HL(-)	P(+)	R(+)	F1(+)
additive	0.0105	0.802	0.779	0.790
C-Block	0.0102	0.820	0.808	0.814
SA-Block	0.0099	0.835	0.810	0.822
H-Decoder	0.0104	0.809	0.791	0.800
Our Model	**0.0093**	**0.860**	**0.843**	**0.851**

model. We also use the evaluation metrics corresponding to the complete model to facilitate this comparison; none of our proposed modules interact with each other; therefore, they can be evaluated independently.

Table 4 shows the contribution of each module to the results on the Zhihu-QT dataset. In addition to the poor performance when not using additive attention, the local neighborhood information extracted by the convolution block contributes greatly to the performance of our model: reducing the Hamming loss by 8.8% and increasing the micro-F_1 score by 4.5%. This result demonstrates the importance of considering the local neighborhood information in the source text when predicting labels. Moreover, our model further demonstrates the ability to extract fine-grained label classification information from the source text.

The hierarchical decoder we designed also provides a large contribution and has a strong influence on model performance. An ordinary decoder aggregates two kinds of fine-grained information from the source text by concatenating vectors. We believe that the method is lossy and does not make full use of the available information. Our model utilizes two independent serial decoding blocks that receive two different types of fine-grained information from the encoder, thus avoiding information loss. Furthermore, any confusion between the two types of information is avoided by using them as input to different decoding blocks. Our method not only preserves the integrity of the two information types but also maintains the weighted independence of label prediction. The hierarchical decoder achieves a Hamming loss reduction of 10.6% and improves the micro-F_1 score by 6.4%.

C. LABEL/SAMPLE RATIO

During the above experiments, we found an interesting variable that we term the label/sample ratio. Our model is highly sensitive to this variable is highly sensitive in our model. For the multi-label text classification task, the label/sample ratio is the average number of labels for all the samples in the entire dataset. To further investigate this variable, we reprocessed the Zhihu-QT dataset. First, we set the label/sample ratio to values roughly equal to 1.2, 1.5, 2.0, 2.5 and 3.0. Then, we selected five subdatasets from the Zhihu-QT based on these fixed values. The number of samples in each subdataset ranged from 35 K to 40 K. Finally, we used the RNN-based Seq2Seq model with attention as the baseline for our model and perform experiments on these five subdatasets.

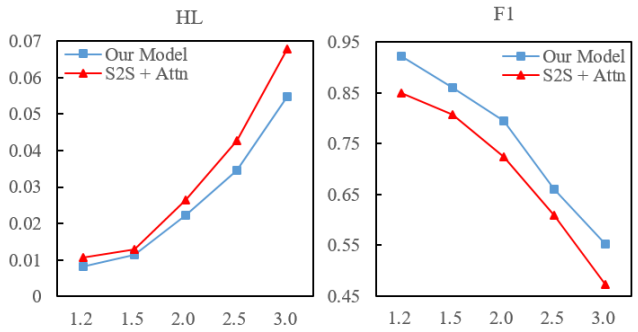


FIGURE 2. Performance of the two models with different label/sample ratio. "S2S + Attn" denotes RNN-based Seq2Seq model with attention.

Figure 2 shows the performances of the two models under different label/sample ratios. Both the RNN-based Seq2Seq model and our models remain at a high level under the initial label/sample ratio. As the label/sample ratio increases, the Hamming loss rises rapidly, and the micro-F_1 score drops sharply. We believe that the performance degradation of the two models is reasonable, because one intuitive explanation is that the more labels a sample contains, the higher the computational complexity of the model are and the higher the performance requirements of the model become. Therefore, under the condition that the model parameters are fixed, the evaluation metrics of the models increase and decrease with the label/sample ratio to varying degrees.

Comparing the two models, we can see that our model performs better than the baseline at each fixed value. We believe that our model is more suitable for high label/sample ratio datasets than the RNN-based Seq2Seq model with attention. Our model captures and efficiently utilizes the fine-grained information in the source text; thus, it outperforms the RNN-based Seq2Seq model on the evaluation metrics.

VII. CONCLUSION AND FUTURE WORK

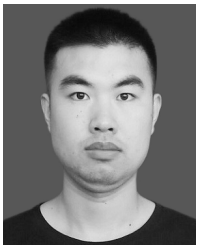
In this study, we proposed a novel sequence-to-sequence learning strategy called "parallel encoding, serial decoding" for multi-label text classification tasks and designed a novel sequence-to-sequence model based on this strategy. The model extracts both fine-gained local neighborhood information and global interaction information from the source text in parallel and utilizes the designed hierarchical decoder for near lossless decoding. The experimental results demonstrate that our proposed model significantly outperforms the current baseline models. Further analysis shows that our model is even more competitive on dataset with high label/sample ratio. However, as shown in Figure 2, our model's performance is still far from ideal on high label/sample ratio datasets. Thus, how to further improve the performance of the model requires more exploration in the future.

REFERENCES

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

[2] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.
[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.
[4] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *J. Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.
[5] J. D. Lafferty, A. Mccallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," Tech. Rep., 2001, pp. 282–289.
[6] C. Li, B. Wang, V. Pavlu, and J. A. Aslam, "Conditional bernoulli mixtures for multi-label classification," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2482–2491.
[7] G. Kurata, B. Xiang, and B. Zhou, "Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jan. 2016, pp. 521–526.
[8] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2017, pp. 2377–2383.
[9] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, "Learning deep latent space for multi-label classification," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2838–2844.
[10] J. Nam, E. L. Mencía, H. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5413–5423.
[11] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[14] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[16] P. Yang, X. Su, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. Int. Conf. Comput. Linguistics*, 2018, pp. 3915–3926.
[17] J. Lin, X. Su, P. Yang, S. Ma, and Q. Su, "Semantic-unit-based dilated convolution for multi-label text classification," in *Proc. Empirical Methods Natural Lang. Process.*, 2018, pp. 4554–4564.
[18] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1746–1751.
[19] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. Meeting Assoc. Comput. Linguistics*, 2014, pp. 655–665.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[21] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. Eur. Conf. Principles Data Mining Knowl. Discovery*, 2001, pp. 42–53.
[22] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 681–687.
[23] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.
[24] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *J. Mach. Learn.*, vol. 73, no. 2, pp. 133–153, Nov. 2008.
[25] N. Ghamrawi and A. Mccallum, "Collective multi-label classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 195–200.
[26] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
[27] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification—Revisiting neural networks," in *Proc. Eur. Conf. Mach. Learn.*, 2014, pp. 437–452.
[28] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[30] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.
[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: https://arxiv.org/abs/1607.06450
[32] S. Wiseman and A. M. Rush, "Sequence-to-sequence learning as beam-search optimization," in *Proc. Empirical Methods Natural Lang. Process.*, 2016, pp. 1296–1306.
[33] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," in *Proc. Conf. Learn. Theory*, 1998, vol. 37, no. 3, pp. 297–336.
[34] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. 2010.
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, Dec. 2014.
[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[37] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1310–1318.
[38] P. Szymański and T. Kajdanowicz, "A scikit-based python environment for performing multi-label classification," 2017, *arXiv:1702.01460*. [Online]. Available: https://arxiv.org/abs/1702.01460

ZHENYU YANG received the M.S. degree in computer application technology from the Qilu University of Technology (Shandong Academy of Sciences), in 2007. He is currently pursuing the Ph.D. degree in control engineering and theory with the China University of Mining and Technology. He is currently an Associate Professor with the Qilu University of Technology (Shandong Academy of Sciences) and the Deputy Director of the Software Integration Institute. His research interests include artificial intelligence, knowledge management, and information integration.



GUOJING LIU received the B.S. degree in measurement and control technology and equipment from the Qilu University of Technology (Shandong Academy of Sciences), in 2017, where he is currently pursuing the M.S. degree with the School of Computer Science and Technology. His research interests include natural language processing, recommendation systems, and image processing.