Label-Specific Document Representation for Multi-Label Text Classification

Lin Xiao, Xin Huang, Boli Chen, Liping Jing
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, China
{17112079, 18120367, 18120345, lpjing}@bjtu.edu.cn

Abstract

Multi-label text classification (MLTC) aims to tag most relevant labels for the given document. In this paper, we propose a Label-Specific Attention Network (**LSAN**) to learn the new document representation. LSAN takes advantage of label semantic information to determine the semantic connection between labels and document for constructing label-specific document representation. Meanwhile, the self-attention mechanism is adopted to identify the label-specific document representation from document content information. In order to seamlessly integrate the above two parts, an adaptive fusion strategy is designed, which can effectively output the comprehensive document representation to build multi-label text classifier. Extensive experimental results on four benchmark datasets demonstrate that LSAN consistently outperforms the state-of-the-art methods, especially on the prediction of low-frequency labels. The code and hyper-parameter settings are released to facilitate other researchers [1].

1 Introduction

Text classification is a fundamental text mining task including multi-class classification and multi-label classification. The former only assigns one label to the given document, while the latter classifies one document into different topics. In this paper, we focus on multi-label text classification (MLTC) because it has become one of the core tasks in natural language processing and has been widely applied in topic recognition (Yang et al., 2016), question answering (Kumar et al., 2016), sentimental analysis (Cambria et al., 2014) and so on. With the boom of big data, MLTC becomes significantly challenging because it has to handle the massive documents, words and labels simultaneously. Therefore, it is an emergency to develop effective multi-label text classifier for various practical applications.

Multi-label text classification allows for the co-existence of more than one label in a single document, thus, there are semantical correlations among labels because they may share the same subsets of document. Meanwhile, the document may be long and complicated semantic information may be hidden in the noisy or redundant content. Furthermore, most documents fall into few labels while a large number of "tail labels" only contain very few positive documents. To handle these issues, researchers pay much attention on three facets: 1) how to sufficiently capture the semantic patterns from the original documents, 2) how to extract the discriminative information related to the corresponding labels from each document, and 3) how to accurately mine the correlation among labels.

Till now, in the community of machine learning and natural language processing, researchers have paid tremendous efforts on developing MLTC methods in each facet. Among them, deep learning-based methods such as CNN (Liu et al., 2017; Kurata et al., 2016), RNN (Liu et al., 2016), combination of CNN and RNN (Lai et al., 2015; Chen et al., 2017), attention mechanism (Yang et al., 2016; You et al., 2018), (Adhikari et al., 2019) and etc., have achieved great success in document representation. However, most of them only focus on document representation but ignore the correlation among labels. Recently, some methods including DXML(Zhang et al., 2018), EXAM(Du et al., 2018), SGM(Yang et al., 2018), GILE(Pappas and Henderson, 2019) are proposed to capture the label correlations by exploiting label structure or label content. Although they obtained promising results in some cases, they still

[1] https://github.com/EMNLP2019LSAN/LSAN/

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 466–475,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

cannot work well when there is no big difference between label texts (e.g., the categories *Management* vs *Management moves* in Reuters News), which makes them hard to distinguish.

In MLTC task, one document may contain multiple labels, and each label can be taken as one aspect or component of the document, thus, the overall semantics of the whole document can be formed by multiple components. Motivated by the above-mentioned observations, we propose a novel **L**abel-**S**pecific **A**ttention **N**etwork model (**LSAN**) to learn document representation by sufficiently exploiting the document content and label content. To capture the label-related component from each document, we adopt the self-attention mechanism to measure the contribution of each word to each label. Meanwhile, **LSAN** takes advantage of label texts to embed each label into a vector like word embedding, so that the semantic relations between document words and labels can be explicitly computed. Thereafter, an adaptive fusion strategy is designed to extract the proper amount of information from these two aspects and construct the label-specific representation for each document. We summarize the main contributions:

- A label-specific attention network model is proposed to handle multi-label text classification task by considering document content and label texts simultaneously.
- An adaptive fusion strategy is first designed to adaptively extract the proper semantical information to construct label-specific document representation.
- The performance of **LSAN** is thoroughly investigated on four widely-used benchmark datasets in terms of several evaluation metrics, indicating its advantage over the state-of-the-art baselines.

The rest of the paper is organized as follows. Section 2 describes the proposed **LSAN** model for multi-label text classification. The experiments on real-word datasets are conducted in Section 3 and their results are discussed in detail. Section 4 lists the related work. The brief conclusions and future work are given in Section 5.

2 Proposed Method

In this section, we introduce the proposed label-specific attention network, as shown in Figure 1. **LSAN** consists of two main parts. The first part is to capture the label-related components from each document by exploiting both document content and label texts. The second part aims to adaptively extract the proper information from two aspects. Finally, the classification model can be trained on the fused label-specific document representations.

2.1 Preliminaries

Problem Definition: Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ denote the set of documents, which consists of N documents with corresponding labels $Y = \{y_i \in \{0,1\}^l\}$, here l is the total number of labels. Each document contains a sequence of words. Each word can be encoded to a low-dimensional space and represented as a d-dimension vector via word2vector technique (Pennington et al., 2014). Let $x_i = \{w_1, \cdots, w_p, \cdots, w_n\}$ denote the i-th document, $w_p \in \mathbb{R}^k$ is the p-th word vector in the document, n is the number of words in document.

For text classification, each label contains textual information. Thus, similar to the document word, one label can be represented as an embedding vector and the label set will be encoded by a trainable matrix $C \in \mathbb{R}^{l \times k}$. Given the input documents and their associated labels D, MLTC aims to train a classifier to assign the most relevant labels to the new coming documents.

Input Text Representation: To capture the forward and backward sides contextual information of each word, we adopt the bidirectional long short-term memory (Bi-LSTM) (Zhou et al., 2016) language model to learn the word embedding for each input document. At time-step p, the hidden state can be updated with the aid of input and $(p-1)$-th step output.

$$\begin{aligned} \overrightarrow{h_p} &= LSTM(\overrightarrow{h_{p-1}}, w_p) \\ \overleftarrow{h_p} &= LSTM(\overleftarrow{h_{p-1}}, w_p) \end{aligned} \tag{1}$$

where w_p is the embedding vector of the p-th word in the corresponding document, and $\overrightarrow{h_p}, \overleftarrow{h_p} \in \mathbb{R}^k$ indicate the forward and backward word context representations respectively. Then, the whole document can be represented by Bi-LSTM as follows.

$$\begin{aligned} H &= (\overrightarrow{H}, \overleftarrow{H}) \\ \overrightarrow{H} &= (\overrightarrow{h_1}, \overrightarrow{h_2}, \cdots, \overrightarrow{h_n}) \\ \overleftarrow{H} &= (\overleftarrow{h_1}, \overleftarrow{h_2}, \cdots, \overleftarrow{h_n}) \end{aligned} \tag{2}$$

In this case, the whole document set can be taken as a matrix $H \in \mathbb{R}^{2k \times n}$.

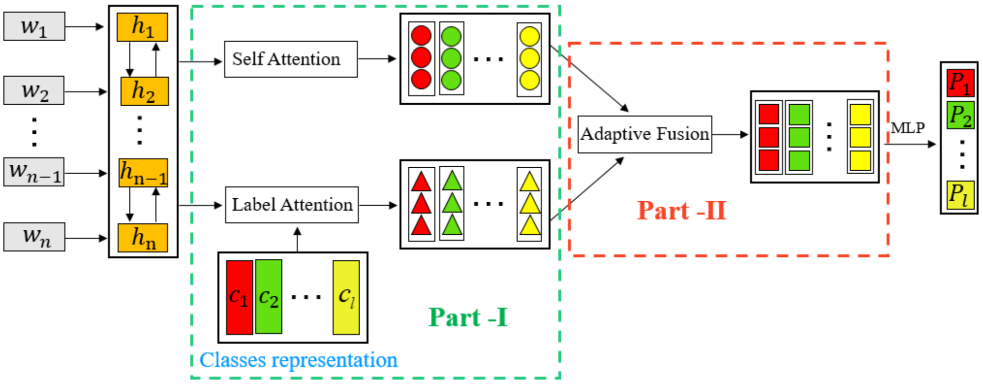


Figure 1: The architecture of the proposed label-specific attention network model (LSAN).

2.2 Label-Specific Attention Network

In this subsection, we will give the proposed attention network model for label-specific document representation learning. It aims to determin the label-related component from each document. Actually, this strategy is intuitive for text classification. For example, regarding the text *"June a friday, in the lawn, a war between the young boys of the football game starte"*, it is assigned into two categories *youth* and *sports*. Obviously, the content *"young boy"* is much more related to *youth* than to *sports*, while *"football game"* should be directly related to *sports*. Next, we will show how to capture this characteristic with our model.

2.2.1 Self-attention Mechanism

As mentioned above, the multi-label document may be tagged by more than one label, and each document should have the most relative contexts with its corresponding labels. In other words, each document may contain multiple components, and the words in one document make different contributions to each label. To capture different components for each label, we adopt the self-attention mechanism (Lin et al., 2017), which has been successful used in various text mining tasks (Tan et al., 2018; Al-Sabahi et al., 2018; You et al., 2018). The label-word attention score ($A^s \in \mathbb{R}^{l \times n}$) can be obtained by

$$A^{(s)} = softmax(W_2 tanh(W_1 H)) \tag{3}$$

where $W_1 \in \mathbb{R}^{d_a \times 2k}$ and $W_2 \in \mathbb{R}^{l \times d_a}$ are the so-called self-attention parameters to be trained. d_a is a hyper-parameter we can set arbitrarily. Each row $A_{j\cdot}^{(s)}$ (an n-dim row vector where n is the total number of words) indicates the contribution of all words to the j-th label. Then, we can obtain the the linear combination of the context words for each label with the aid of label-word attention score ($A^{(s)}$) as follows.

$$M_{j\cdot}^{(s)} = A_{j\cdot}^{(s)} H^T \tag{4}$$

which can be taken as a new representation of the input document along the j-th label. Then the whole matrix $M^{(s)} \in \mathbb{R}^{l \times 2k}$ is the label-specific document representation under the self-attention mechanism.

2.2.2 Label-Attention Mechanism

Self-attention mechanism can be taken as content-based attention because it only considers the document content. As we all know, labels have specific semantics in text classification, which is hidden in the label texts or descriptions. To make use of the semantic information of labels, they are preprocessed and represented as a trainable matrix $C \in \mathbb{R}^{l \times k}$ in the same latent k-dim space with the words.

Once having the word embedding from Bi-LSTM in (1) and the label embedding in C, we can explicitly determine the semantic relation between each pair of word and label. A simple way is calculating the dot product between $\overrightarrow{h}_p$ (or $\overleftarrow{h}_p$) and $C_{j\cdot}$ as follows.

$$\begin{aligned} \overrightarrow{A}^{(l)} &= C\overrightarrow{H} \\ \overleftarrow{A}^{(l)} &= C\overleftarrow{H} \end{aligned} \tag{5}$$

where $\overrightarrow{A}^{(l)} \in \mathbb{R}^{l \times n}$ and $\overleftarrow{A}^{(l)} \in \mathbb{R}^{l \times n}$ indicate the forward and backward sides semantic relation between words and labels. Similar to the previous self-attention mechanism, the label-specific document representation can be constructed by linear combining the label's context words as follows.

$$\begin{aligned} \overrightarrow{M}^{(l)} &= \overrightarrow{A}^{(l)}\overrightarrow{H}^T \\ \overleftarrow{M}^{(l)} &= \overleftarrow{A}^{(l)}\overleftarrow{H}^T \end{aligned} \tag{6}$$

Finally, the document can be re-represented along all labels via $M^{(l)} = (\overrightarrow{M}^{(l)}, \overleftarrow{M}^{(l)}) \in \mathbb{R}^{l\times 2k}$. This representation is based on the label texts, thus, we called it as label-attention mechanism.

2.3 Adaptive Attention Fusion Strategy

Both $M^{(s)}$ and $M^{(l)}$ are label-specific document representation, but they are different. The former focuses on the document content, while the later prefers to the semantic correlation between document content and label text. In order to take advantage these two parts, in this subsection, an attention fusion strategy is proposed to adaptively extract proper amount of information from them and build comprehensive label-specific document representation.

More specifically, two weight vectors ($\alpha, \beta \in \mathbb{R}^l$) are introduced to determine the importances of the above two mechanisms, which can obtained by a fully connected layer on the input $M^{(s)}$ and $M^{(l)}$.

$$\begin{aligned} \alpha &= sigmoid(M^{(s)}W_3) \\ \beta &= sigmoid(M^{(l)}W_4) \end{aligned} \tag{7}$$

$W_3, W_4 \in \mathbb{R}^{2k}$ are the parameters to be trained. α_j and β_j indicate the importances of self-attention and label-attention to construct the final document representation along the j-th label respectively. Therefore, we add the constraint on them:

$$\alpha_j + \beta_j = 1 \tag{8}$$

Then, we can obtain the final document representation along the j-th label based on fusion weights as follows.

$$M_{j\cdot} = \alpha_j M^{(s)}_{j\cdot} + \beta_j M^{(l)}_{j\cdot} \tag{9}$$

The label-specific document representation along all labels can be described as a matrix $M \in \mathbb{R}^{l\times 2k}$.

2.4 Label Prediction

Once having the comprehensive label-specific document representation, we can build the multi-label text classifier via a multilayer perceptron with two fully connected layers. Mathematically, the predicted probability of each label for the coming document can be estimated via

$$\hat{y} = sigmoid(W_6 f(W_5 M^T)) \tag{10}$$

Here $W_5 \in \mathbb{R}^{b\times 2k}, W_6 \in \mathbb{R}^b$ are the trainable parameters of the fully connected layer and output layer respectively. f is the ReLU nonlinear activation function. The sigmoid function is used to transfer the output value into a probability, in this case, the cross-entropy loss can be used as the loss function which has been proved suitable for multi-label text classification task (Nam et al., 2014).

$$\begin{aligned} \mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{l}&(y_{ij}\log(\hat{y}_{ij})) \\ &+ (1-y_{ij})\log(1-\hat{y}_{ij}) \end{aligned} \tag{11}$$

where N is the number of training documents, l is the number of labels, $\hat{y}_{ij} \in [0, 1]$ is the predicted probability, and $y_{ij} \in \{0, 1\}$ indicates the ground truth of the i-th document along the j-th label.

3 Experiments

In this section, we evaluate the proposed model on four datasets (with various number of labels from 54 to 3956) by comparing with the state-of-the-art methods in terms of widely used metrics.

3.1 Experimental Setting

Datasets: There are several multi-label text datasets, however, only few of them have label text information. Thus, in this paper, four benchmark multi-label datasets including three small-scale datasets (RCV1, AAPD and EUR-Lex) and one medium-scale dataset (KanShan-Cup) are used to construct the experiments.

- **Reuters Corpus Volume I (RCV1)** (Lewis et al., 2004) contains more than 80K manually categorized news belonging to 103 classes.
- **AAPD**[2] (Yang et al., 2018) collects the abstract and the corresponding subjects of 55840 papers from arXiv in the filed of computer science.
- **EUR-Lex** (Mencia and Fürnkranz, 2008) is a collection of documents about European Union law belonging to 3956 subjects. The public version[3] contains 11585 training instances and 3865 testing instances.
- **KanShan-Cup**[4] is released by the largest Chinese community question answering platform, Zhihu. It contains near 3 million questions about 1999 topics.

For the first three data sets, only last 500 words were kept for each document, while the last 50

[2] https://github.com/lancopku/SGM

[3] https://drive.google.com/drive/folders/1KQMBZgACUm-ZZcSrQpDPlB6CFKvf9Gfb

[4] https://www.biendata.com/competition/zhihu/data/

Table 1: Summary of Experimental Datasets.

Datasets	N	M	D	L	$\bar{L}$	$\tilde{L}$	$\bar{W}$	$\tilde{W}$
RCV1	23,149	781,265	47,236	103	3.18	729.67	259.47	269.23
AAPD	54,840	1,000	69,399	54	2.41	2444.04	163.42	171.65
EUR-Lex	11,585	3,865	171,120	3,956	5.32	15.59	1,225.20	1,248.07
Kanshan-Cup	2,799,967	200,000	411,721	1999	2.34	3513.13	38.06	35.48

N is the number of training instances, M is the number of test instances, D is the total number of words, L is the total number of classes, $\bar{L}$ is the average number of labels per document, $\tilde{L}$ is the average number of documents per label, $\bar{W}$ is the average number of words per document in the training set, $\tilde{W}$ is the average number of words per document in the testing set.

words were used in KanShan-Cup dataset. Once the document has less than the predifined number of words, we extend it by padding zeros. All methods are trained and tested on the given training and testing datasets which are summarized in Table 1.

Evaluation Metrics: We use two kinds of metric, precision at top K ($P@k$) and the Normalized Discounted Cumulated Gains at top K ($nDCG@k$) to evaluate the prediction performance. $P@k$ and $nDCG@k$ are defined according to the predicted score vector $\hat{y} \in \mathbb{R}^l$ and the ground truth label vector $y \in \{0,1\}^l$ as follows.

$$P@k = \frac{1}{k} \sum_{l \in rank_k(\hat{y})} y_l$$

$$DCG@k = \sum_{l \in rank_k(\hat{y})} \frac{y_l}{\log(l+1)}$$

$$nDCG@k = \frac{DCG@k}{\sum_{l=1}^{\min(k,\|y\|_0)} \frac{1}{\log(l+1)}}$$

where $rank_k(y)$ is the label indexes of the top k highest scores of the current prediction result. $\|y\|_0$ counts the number of relevant labels in the ground truth label vector y.

Baseline Models: The proposed LSAN is a deep neural network model, thus the recent state-of-the-art deep learning-based MLTC methods are selected as baselines.

- **XML-CNN:** (Liu et al., 2017) adopts Convolutional Neural Network (CNN) and a dynamic pooling technique to extract high-level feature for multi-label text classification.
- **SGM:** (Yang et al., 2018) applies a sequence generation model from input document to output label to construct the multi-label text classifier.
- **DXML:** (Zhang et al., 2018) tries to explore the label correlation by considering the label structure from the label co-occurrence graph.
- **AttentionXML:** (You et al., 2018) builds the label-aware document representation only based on the document content, thus, it can be taken as one special case of our proposed **LSAN** with arbitrarily setting $\alpha = 0$.
- **EXAM:** (Du et al., 2018) is the most similar work to **LSAN** because both of them exploit the label text to learn the interaction between words and labels. However, EXAM suffers from the situation where different labels have similar text.

Parameter Setting: For the KanShan-Cup dataset, we use the pre-trained word embedding and label embedding public in the official website, where the embedding space size is 256, i.e., $k = 256$. The parameters corresponding to the weights between neurals are $d_a = 200$ for W_1 and W_2, $b = 256$ for W_5 and W_6. For other three datasets, $k = 300$, $d_a = 200$ and $b = 300$. The whole model is trained via Adam (Kingma and Ba, 2014) with the learning rate being 0.001. The parameters of all baselines are either adopted from their original papers or determined by experiments.

3.2 Comparison Results and Discussion

In this section, the proposed LSAN is evaluated on four benchmark datasets by comparing with five baselines in terms of $P@K$ and $nDCG@K (K = 1, 3, 5)$. Table 2 and Table 3 show the averaged performance of all test documents. According to the formula of $P@K$ and $nDCG@K$, we know $P@1 = nDCG@1$, thus only $nDCG@3$ and $nDCG@5$ are listed in Table 3. In each line, the best result is marked in bold.

From Table 2 and 3, we can make a number of observations about these results. Firstly, XML-CNN is worse than other four methods because it only considers the document content but ignores the label correlation which has been proven very important for multi-label learning. Secondly, AttentionXML is superior to EXAM on datasets RCV1 and Kanshan-Cup, because these two datases have hierarchical label structures. In this case, parent label and child label may contain similar text, which makes them hard to distinguish according to the text-based embedding and further reduce the performance of EXAM. By compar-

Datasets	Metrics	XML-CNN	DXML	SGM	AttentionXML	EXAM	LSAN(ours)
RCV1	P@1	95.75%	94.04%	95.37%	96.41%	93.67%	**96.81%**
	P@3	78.63%	78.65%	81.36%	80.91%	75.80%	**81.89%**
	P@5	54.94%	54.38%	53.06%	56.38%	52.73%	**56.92%**
AAPD	P@1	74.38%	80.54%	75.67%	83.02%	83.26%	**85.28%**
	P@3	53.84%	56.30%	56.75%	58.72%	59.77%	**61.12%**
	P@5	37.79%	39.16%	35.65%	40.56%	40.66%	**41.84%**
EUR-Lex	P@1	70.40%	75.53%	70.45%	67.34%	74.40%	**79.17%**
	P@3	54.98%	60.13%	60.37%	52.52%	61.93%	**64.99%**
	P@5	44.86%	48.65%	43.88%	47.72%	50.98%	**53.67%**
Kanshan-Cup	P@1	49.68%	50.84%	50.32%	53.69%	51.41%	**54.46%**
	P@3	32.27%	32.69%	31.83%	34.10%	32.81%	**34.56%**
	P@5	24.17%	24.07%	23.95%	25.16%	24.29%	**25.54%**

Table 2: Comparing LSAN with five baselines in terms of $P@K$ (K=1,3,5)on four benchmark datasets.

Datasets	Metrics	XML-CNN	DXML	SGM	AttentionXML	EXAM	LSAN(ours)
RCV1	nDCG@3	89.89%	89.83%	91.76%	91.88%	86.85%	**92.83%**
	nDCG@5	90.77%	90.21%	90.69%	92.70%	87.71%	**93.43%**
AAPD	nDCG@3	71.12%	77.23%	72.36%	78.01%	79.10%	**80.84%**
	nDCG@5	75.93%	80.99%	75.35%	82.31%	82.79%	**84.78%**
EUR-Lex	nDCG@3	58.62%	63.96%	60.72%	56.21%	65.12%	**68.32%**
	nDCG@5	53.10%	57.60%	55.24%	50.78%	59.43%	**62.47%**
Kanshan-Cup	nDCG@3	46.65%	49.54%	46.90%	51.03%	49.32%	**51.43%**
	nDCG@5	49.60%	52.16%	50.47%	53.96%	49.74%	**54.36%**

Table 3: Comparing LSAN with five baselines in terms of $nDCG@K$ (K=3,5) on four benchmark datasets.

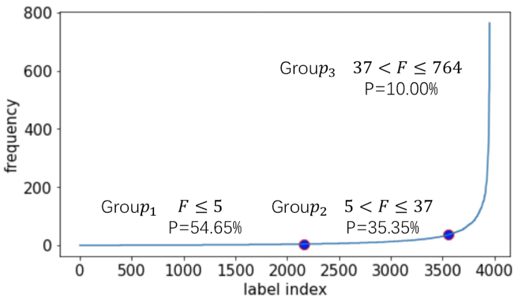


Figure 2: The label distribution of EUR-Lex

ing with EXAM and the proposed LSAN, however, AttentionXML performs worse on EUR-Lex dataset The main reason is that AttentionXML only focuses on the document content, which will make it not sufficiently trained once there are only few documents in some labels. Fortunately, EXAM and LSAN benefit from the label texts. Last one, as expected, is that LSAN consistently outperforms all baselines on all experimental datasets. This result further confirms that the proposed adaptive attention fusion strategy is much helpful to learn the label-specific document representation for multi-label text classification.

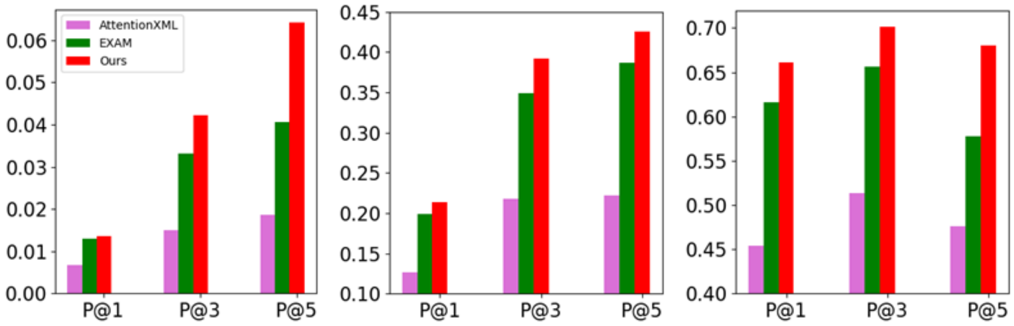
3.3 Comparison on Sparse Data

In order to verify the performance of LSAN on low-frequency labels, we divided labels in EUR-Lex into three groups according to their occurring frequency. Figure 2 shows the distribution of label frequency on EUR-Lex, F is the frequency of label. Among it, nearly 55% of labels occur between 1 and 5 times to form the first label group (Group1). The labels appearing 5-37 times are assigned into Group2, which is 35% of the whole label set. The remaining 10% frequent labels form the last group (Group3). Obviously, Group1 is much harder than other two groups due to the lack of training documents.

Figure 3 shows the prediction results in terms of $P@1$, $P@3$ and $P@5$ obtained by AttentionXML, EXAM and LSAN. Three methods become better and better from Group1 to Group3, which is reasonable because more and more documents are included in each label from Group1 to Group3. LSAN significantly improves the prediction performance on Group1. Especially, LSAN obtains an average of more than 83.82%, 182.55%, 244.62% gain on three metrices for group 1 to AttentionXML, and 3.85%, 27.19%, 58.27% gain to EXAM. This result demonstrates the superiority of the proposed model on multi-label text data with tail labels.

3.4 Ablation Test

The proposed LSAN can be taken as a joint attention strategy including three parts. One is self-attention based on document content (denoted as A). The second one is label-attention based on label text (denoted as L). Another one is fusion attention by adaptively integrating A and L with proper weights (denoted as W). In this section, we

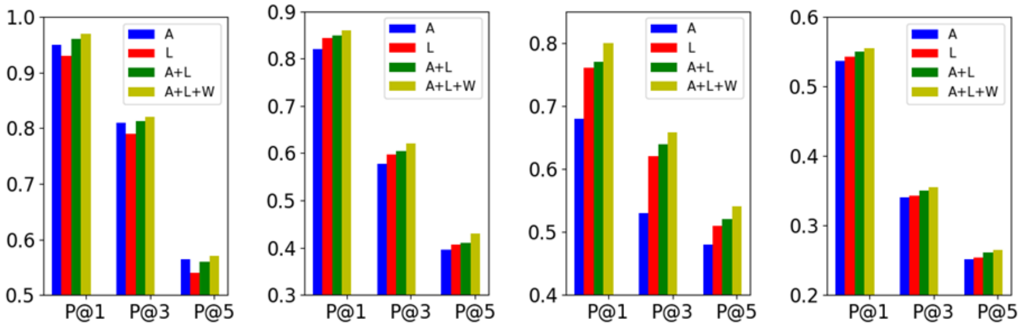


(a) Gropu1 ($F \leq 5$)

(b) Gropu2 ($5 < F \leq 37$)

(c) Gropu3 ($F > 37$)

Figure 3: Precision@k for three groups on EUR-Lex



(a) RCV1

(b) AAPD

(c) EUR-Lex

(d) Kanshan-Cup

Figure 4: Result of the ablation test. ‘A’ denotes the Self-Attention, ‘L’ denotes the Label-Attention, ‘W’ denotes the Fusion Attention with Adaptive Weights.

try to demonstrate the effection of each component via an ablation test.

Figure 4 lists the prediction results on four datasets in terms of $P@1, P@3$ and $P@5$. (i.e., S+L gets better results than S and L). S prefers to finding the useful content when constructing the label-specific document representation, but it ignores the label information. L takes adantage of label text to explicitly determine the semantic relation between documents and labels, however, label text is not easy to distinguish the difference between labels (e.g., *Management* vs. *Management movies*). Therefore, coupling with both attentions is really reasonable. Furthermore, adaptively extracting proper amount of information from these two attentions benefits the final multi-label text classification.

To further verify the effectiveness of attention adaptive fusion, Figure 5 lists the distribution of weights on Self-attention and Label-attention on two representative datasets, one for sparse data (EUR-Lex) and the other for dense data (AAPD). As expected, the label-attention is much more useful than self-attention for sparse data, vice versa for dense data. In dense data, each label has sufficient documents, therefore, self-attention can sufficiently obtain label-specific document representation. On the other hand, label text is helpful to extract the semantic relations between labels and documents. Results on other two datasets have the similar trends which are omitted due to page limitation.

For investigating the effect of label-attention, we visualize the attention weights on the original document using heat map, as shown in Figure 6. Among it, the example AAPD document belongs to two categories *Computer Vision* and *Neural and Evolutionary Computing*. From the attention weights, we can see that each category has its own most related words, which confirms that the proposed label specific attention network is able to extract the label-aware content and further construct label-specific document representation.

4 Related work

In the line of MLTC, most works focus on two issues, one is document representation learning and

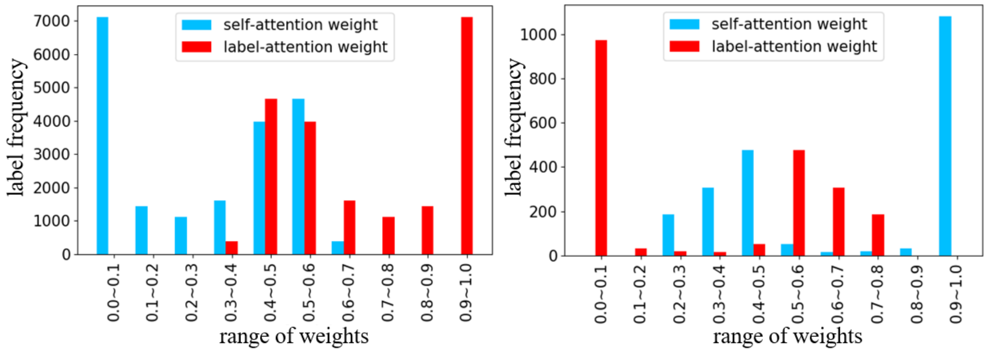
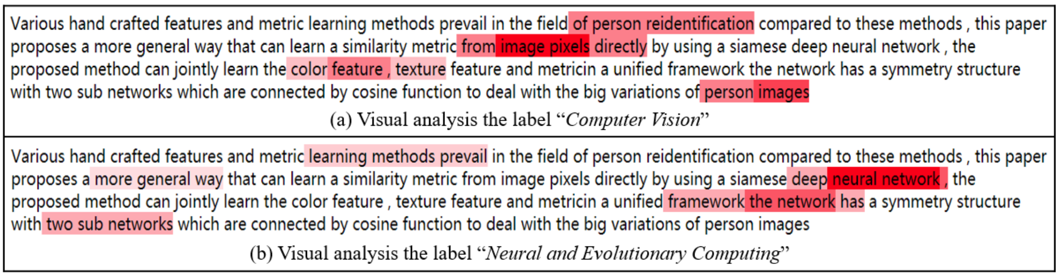


Figure 5: Weight distribution for two components on EUR-Lex (left subfigure) and AAPD (right subfigure). Horizontal axis is the range of weight from 0 to 1 with 0.1 gap. Vertical axis is the frequency that the specific range occurs in current label group.



Various hand crafted features and metric learning methods prevail in the field of person reidentification compared to these methods , this paper proposes a more general way that can learn a similarity metric from image pixels directly by using a siamese deep neural network , the proposed method can jointly learn the color feature , texture feature and metricin a unified framework the network has a symmetry structure with two sub networks which are connected by cosine function to deal with the big variations of person images

(a) Visual analysis the label "*Computer Vision*"

Various hand crafted features and metric learning methods prevail in the field of person reidentification compared to these methods , this paper proposes a more general way that can learn a similarity metric from image pixels directly by using a siamese deep neural network , the proposed method can jointly learn the color feature , texture feature and metricin a unified framework the network has a symmetry structure with two sub networks which are connected by cosine function to deal with the big variations of person images

(b) Visual analysis the label "*Neural and Evolutionary Computing*"

Figure 6: Demonstration of words with largest label-attention weights ($A^{(l)}$) in one AAPD document belonging to two categories:*Computer Vision and Neural and Evolutionary Computing*.

the other is label correlation detection.

For document representation, along with the recent success in CNN, many works are proposed based on CNN (Kim, 2014; Liu et al., 2017; Chen et al., 2017), which can capture the local correlations from the consecutive context windows. Although they obain promising results, these methods suffer from the limitation of window size so that they cannot determine the long-distance dependency of text. Meanwhile, they treat all words equally no matter how noisy the word is. Later, RNN and attention mechanism are introduced to get brilliant results (Yang et al., 2016). To implicitly learn the document representation for each label, the self-attention mechanism (Lin et al., 2017) is adopted for multi-label classification (You et al., 2018).

To determine the label correlation among multi-label data, in literatures, researchers proposed various methods. Kurata et al. (2016) adopt an initialization method to leverage label co-occurrence information. SLEEC (Bhatia et al., 2015) divides dataset into several clusters, and in each cluster it detects embedding vectors by capturing non-linear label correlation. DXML (Zhang et al., 2018) establishes an explicit label co-occurrence graph to explore label embedding in low-dimension latent space. Yang et al. (2018) use sequence-to-sequence(Seq2Seq) model to consider the correlations between labels. Recently, the textual information of labels are used to guide MLTC. EXAM (Du et al., 2018) introduces the interaction mechanism to incorporate word-level matching signals into the text classification task. GILE (Pappas and Henderson, 2019) proposes a joint input-label embedding model for neural text classification. Unfortunately, they cannot work well when there is no big difference between label texts.

5 Conclusions and Future Work

A new label-specific attention network, in this paper, is proposed for multi-label text classification. It makes use of document content and label text to learn the label-specific document representation with the aid of self-attention and label-attention mechanisms. An adaptive fusion is designed to effectively integrate these two attention mechanisms to improve the final prediction performance. Extensive experiments on four benchmark datasets prove the superiority of LSAN by comparing with the state-of-the-art methods, especially on the dataset with large subset of low-frequency labels.

In real applications, more precious information can be collected, such as label description, label topology (e.g., hierarchical structure) and etc. Therefore, it is interesting to extend the current model with such extra information.

Acknowledgments

We would like to thank the outstanding anonymous reviewers for their helpful comments to improve our manuscript. This work was supported in part by the National Natural Science Foundation of China under Grant 61822601, 61773050, and 61632004; the Beijing Natural Science Foundation under Grant Z180006; the Beijing Municipal Science & Technology Commission under Grant Z181100008918012; National Key Research and Development Program (2017YFC1703506).

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in neural information processing systems*, pages 730–738.

Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. 2014. Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*.

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2377–2383. IEEE.

Cunxiao Du, Zhaozheng Chin, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2018. Explicit interaction model towards text classification. *arXiv preprint arXiv:1811.09386*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Improved neural network-based multi-label classification with better initialization leveraging label cooccurrence. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 521–526.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multilabel text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Eneldo Loza Mencia and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for largescale problems in the legal domain. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 50–65. Springer.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Largescale multi-label text classificationłrevisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer.

Nikolaos Pappas and James Henderson. 2019. Gile: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics (TACL)*, 7.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, and Shanfeng Zhu. 2018. Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks. *arXiv preprint arXiv:1811.01727*.

Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 100–107. ACM.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.