Label-aware Document Representation via Hybrid Attention for Extreme Multi-Label Text Classification

Xin Huang¹, Boli Chen¹, Lin Xiao¹, and Liping Jing¹

Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China
{18120367, 18120345, 17112079, lpjing}@bjtu.edu.cn

Abstract. Extreme multi-label text classification (XMTC) aims at tagging a document with the most relevant labels from an extremely large-scale label set. It is a challenging problem especially for the tail labels because there are only few training documents to build classifier. This paper is motivated to better explore the semantic relationship between each document and extreme labels by taking advantage of both document content and label correlation. Our objective is to establish an explicit label-aware representation for each document with a hybrid attention deep neural network model (LAHA). LAHA consists of three parts. The first part adopts a multi-label self-attention mechanism to detect the contribution of each word to labels. The second part exploits the label structure and document content to determine the semantic connection between words and labels in a same latent space. An adaptive fusion strategy is designed in the third part to obtain the final label-aware document representation so that the essence of previous two parts can be sufficiently integrated. Extensive experiments have been conducted on six benchmark datasets by comparing with the state-of-the-art methods. The results show the superiority of our proposed LAHA method, especially on the tail labels.

Keywords: Extreme Multi-label Text Classification · Deep Neural Network · Attention · Tail Label · Lable-aware Document Representation.

1 Introduction

Extreme multi-label text classification (XMTC) aims at automatically tagging a document with the most relevant labels from an extremely large label set. For instance, there are millions of categories on Wikipedia and one might wish to build a classifier that can annotate a given message with the subset of most relevant Wikipedia categories [2]. XMTC has become increasingly important due to the boom of big data, while it becomes significantly challenging because it has to simultaneously handle massive documents, features and labels. Thus it is emergency to develop effective extreme multi-label classifier for various real-applications such as product categorization in e-commerce, news annotation and etc.
Multi-label text classification, unlike the traditional multi-class classification, allows for the co-existence of more than one labels for a single document. Meanwhile, there may be a large number of 'tail labels' with very few positive documents in XMTC tasks. To tackle the aforementioned issues, researchers pay much attention on two facets: 1) how to represent label so that the correlation among labels can be accurately mined, and 2) how to represent document so that the dependency among text can be sufficiently captured. Recently, state-of-the-art extreme multi-label learning methods have been proposed in each facet. Among them, tree-based and embedding-based methods become popular to find the label correlation as they can obtain notable accuracy improvement by constructing a hierarchy structure [18] or learning a low-dimensional latent space [2]. Deep learning-based methods (e.g., convolutional neural network [12]) have achieved great success to represent text data. These methods usually characterize one document with the same representation on all labels. In this case, the probability of document belonging to a class is determined by their overall matching score regardless of the label-aware semantic information. Recent works, AttentionXML [25] and EXAM [6], turn attention to this issue with the aid of attentive neural network. However, they only focus on document or label content but ignoring the label structure among extreme labels which has been proved very important in extreme multi-label learning [2].

To solve the above-mentioned problems, we introduce a Label-Aware document representation model via a Hybrid Attention neural network (LAHA) by considering both document content and label structure. LAHA consists of three parts. The first part aims at detecting the importance of each word to all labels via a self-attention bidirectional LSTM neural network. The second part tries to explore the semantic connection between words and labels in a latent space. Here the word embedding is obtained by the bidirectional LSTM neural network. The label embedding is determined from the label co-exist graph so that the label structure can be sufficiently maintained in the same latent space with words’. Based on these two embeddings, we introduce an interaction-attention mechanism to explicitly compute the semantic relation between the words and labels. The last part is to represent each document along each label via an adaptive fusion strategy. The goal of fusion strategy is to adaptively extract proper information from the previous two parts so that the final document representation has discriminative ability to construct classifier.

The proposed XMTC model LAHA has been evaluated on six benchmark datasets by comparing with the state-of-the-art methods. The results show that LAHA significantly improves the XMTC performance especially for the tail labels (Note that tail label only contains very few training documents which makes the prediction more challenging). We summarize the major contributions.

- LAHA is the first work to construct label-aware document representation by simultaneously considering document content and label structure.
- The hybrid attention mechanism is firstly designed to adaptively extract the semantic relation between each document and all labels for XMTC.
– The performance of LAHA was thoroughly investigated on widely-used benchmark datasets in terms of several evaluation metrics, indicating the advantage over the baselines especially on tail labels.
– The code and hyper-parameter settings are released\(^1\) to facilitate other researchers.

2 Related Work

Significant progress has been made for XMTC. Most methods focus on detecting the label correlation and reducing the computational complexity in the huge-size label space. They can be roughly categorized into two categories: embedding-based and tree-based methods. Recently, due to the powerful ability of representation, deep learning technology has been introduced to effectively represent document for XMTC tasks. Next, we will briefly review them.

2.1 Embedding-based Methods

Embedding-based methods aim at reducing the huge label space to a low dimensional space while preserving the label correlation as much as possible. Various approaches have been presented such as compressed sensing \(8\), output codes \(28\), Singular Value Decomposition \(20\), landmark labels \(1\), Bloom filters \(4\), etc. To efficiently handle large-scale label set, these embedding-based methods usually assume that the label matrix is low-rank. However, such methods have been proved unable to deliver high prediction accuracies as the low rank assumption is violated in most real world applications \(2\). SLEEC \(2\) can be taken as the most representative embedding-based method due to its significant accuracy and computationally efficiency. Its main idea is to learn a small ensemble of local distance preserving embeddings. Specifically, SLEEC divides the training data set into several clusters, and in each cluster it detects embedding vectors by capturing non-linear label correlation and preserving the pairwise distance between labels. The \(k\)-nearest neighbors search is used to do prediction only in the cluster into which the test document is fallen. Later, Zhang et al. \(26\) adopted deep neural network for non-linear modeling the label embedding.

Once having the low-dimensional label embedding, any classifier can be trained from the feature space to the compressed space. As for the prediction stage, labels for a novel document are determined through a lifting decompression matrix, which projects the embedded label vectors back to the original extremely high-dimensional label space. Although these methods perform well, they play a heavy price in terms of prediction accuracy due to the loss of information during the compression and decompression phases.

2.2 Tree-based Methods

Tree-based methods, similar to the traditional Decision-Tree, introduce a tree structure to divide the documents recursively at each non-leaf node, so that

\(^1\) https://github.com/HX-idiot/Hybrid_Attention_XML
documents in each leaf node share the similar label distribution. The most representative method FastXML [18] implements this process by optimizing the normalized discounted cumulative gain (nDCG)-based ranking loss function. Then, a base classifier is trained at each leaf node which only focuses on a few active labels. To enhance the robustness of predictions, an ensemble of multiple induced trees are learned. The main advantage of tree-based methods is that the prediction time complexity is typically sublinear in the training-set size and would be logarithmic if the induced tree is balanced.

A recent extension work of FastXML is PfastreXML [9], which adopted a propensity scored objective function instead of nDCG-based loss. PfastreXML can prioritize predicting the few relevant labels over the large number of irrelevant ones and thus promote the accurate prediction of tail labels. Parabel [17] is another tree-based method, which constructs balanced trees partitioning labels rather than instances. These tree-based methods sufficiently consider the label distribution when training the classifier, but they represent document via bag-of-words, where the words are treated as independent features, which will ignore the semantic dependency among words.

2.3 Deep Learning-based Methods

To solve the fundamental limitation of the aforementioned methods, ignoring the semantic dependency among words, researchers adopted deep learning models in text classification task due to its strong ability of representation. Its main idea is to compress the input word-level representation into a document-level representation, i.e., they encode one document in a latent space [15] by considering the contextual information. The popular deep models include convolutional neural network (CNN) [27], gated recurrent units (GRU) [3], recurrent neural network (RNN) [19], long short-term memory (LSTM) [22], bidirectional LSTM (Bi-LSTM) [30], bidirectional encoder representation from transformer (BERT) [5] and several combination networks [29,23]. Even though they have achieved great success in traditional NLP tasks, few work is designed for XMTC.

XML-CNN [12] can be taken as the first and most representative work using deep learning model in XMTC. It takes advantage of CNN, dynamic max-pooling and bottle-neck layer to build the deep model. Due to the limited window size, XML-CNN can not capture the long-distance dependency among text. Later, GRU and Bi-LSTM language models are adopted in AttentionXML [25] and EXAM [6] to effectively represent document for XMTC. Meanwhile, these two methods consider the difference of one document representation along different labels. AttentionXML [25] adopts self-attention mechanism [11], while EXAM [6] exploits the label content information to calculate the relations between words and classes. Although AttentionXML obtains promising performance, it ignores the label structure which has been proved very important in embedding-based and tree-based multi-label learning methods.

Therefore, in this paper, we propose a new XMTC deep model with hybrid attention to build label-aware document representation, which sufficiently exploits both document content and label structure.
3 LAHA model

In this section, we introduce the proposed deep model (LAHA) to handle XMTC tasks. Our goal is to build a multi-label learning model from the training documents with a large-size label set. Let $D = \{(x_1, y_1), ..., (x_N, y_N)\}$ be the given raw training document set containing total $N$ documents and belonging to $k$ labels. Each document has $n$ tokens (or words) and each word is represented via a $d$-dimensional deep semantic dense vector acquired from word2vec technique, $e_t \in \mathbb{R}^d$ ($t = 1, ..., n$). $y_i \subseteq \{0, 1\}^k$ is the corresponding label vector, and $y_{ij} = 1$ iff the $j$-th label is turned on for the $i$-th document $x_i$.

LAHA mainly consists of three parts: a self-attention Bi-LSTM (as shown in the solid box of Fig. 1) to represent document by considering the contribution of each word to labels, an interaction-attention network (as shown in the dashed box of Fig. 1) to represent document by determining the semantic connection between words and labels, and an attention fusion layer (as shown in the dotted box of Fig. 1) to incorporate the previous parts. The fully connected and output layer is on the top of LAHA to get the label prediction of each input document.

3.1 Feature Embedding

To build the proposed LAHA multi-label text classifier, the raw text data is preprocessed via word embedding technique so that each word is represented as a low-dimensional dense vector. The extreme labels are embedded into dense vectors from the label co-exist graph so that the label correlation and local structure can be sufficiently captured.
Word Embedding Once having the $d$-dimensional word vector $e_t \in \mathbb{R}^d$ for each word ($t = 1, \ldots, n$), the whole document can be taken as a sequence of words $(e_1, \ldots, e_n)$ as the input of LAHA. To capture the forward and backward sides contextual information of each word, we adopt here Bidirectional Long short-term memory (Bi-LSTM) language model [30] to learn the word embedding for each input document. At time-step $t$, the memory $c_t$ and the hidden state $h_t$ can be updated with the aid of input gate, forget gate and output gate. Finally, each word is represented by a new dense vector $h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$, where $\overrightarrow{h}_t \in \mathbb{R}^r$ and $\overleftarrow{h}_t \in \mathbb{R}^r$ are the forward and backward word context representations respectively.

Then, the whole output of Bi-LSTM can be obtained by

$$H = (H^{(f)}; H^{(b)}) \quad \text{with}$$

$$H^{(f)} = ([\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_n]) \in \mathbb{R}^{r \times n}$$

$$H^{(b)} = ([\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_n]) \in \mathbb{R}^{r \times n}$$

In this case, we can say each word is re-represented in the new $r$-dimensional latent space, and the whole document is taken as a matrix $H \in \mathbb{R}^{2r \times n}$.

Label Embedding To better extract label correlation information, we firstly build a label co-exist graph $G = (V, S)$ from the training data. In this graph, $V$ is composed by the label node set. There will be an edge connecting the $i$-th label and the $j$-th label if they share at least one document [26]. Our goal is to represent the extreme labels in a low-dimensional latent space so that two nearby labels in the graph have similar representation, i.e., the local structure among labels are preserved as much as possible. Thus, the popular and powerful node2vec [7] is adopted here because it has ability to explore the labels’ diverse neighborhoods by a flexible biased random walk procedure in a breadth-first sampling as well as depth-first sampling fashion. Finally, each label will be represented by a $r$-dimensional dense vector, i.e., $l_i \in \mathbb{R}^r$ for the $i$-th label ($i = 1, \ldots, k$) and the whole label set can be described by

$$L = (l_1, l_2, \ldots, l_k) \in \mathbb{R}^{r \times k}.$$  

3.2 Hybrid Attention Mechanism

Hybrid attention mechanism aims at better representing each document by taking advantage of both document content and label structure. It is composed of self-attention mechanism on document content and interaction-attention mechanism to exploit document content and label structure.

Self-attention (SA) has been successful used in text mining tasks such as relation extraction [21]. In multi-label data, since one document may be tagged by more than one labels, each document should have the most relative context to its corresponding labels. That is, the words in one document make different
contributions to each label. To focus on different aspects of document, thus, we introduce self-attention mechanism (SA) \[11\] on the output of Bi-LSTM \((H)\). The attention score \(A^{(S)} \in \mathbb{R}^{n \times k}\) is calculated by

\[
A^{(S)} = \text{softmax}(W_{s2}\tanh(W_{s1}H))
\]

(5)

where \(W_{s1} \in \mathbb{R}^{d_a \times 2r}\) and \(W_{s2} \in \mathbb{R}^{k \times d_a}\) are parameters to be trained. The softmax here is performed on first dimension to show different contributions of words for each label. Then, we can obtain the linear combination of context words for each label through self-attention mechanism as follows.

\[
C^{(S)}_j = HA^{(S)}_j
\]

(6)

which can be taken as the representation of the input document along the \(j\)-th label. The whole matrix \(C^{(S)} \in \mathbb{R}^{2r \times k}\) is the label-aware document representation under the self-attention mechanism.

**Interaction-attention (IA)** aims to determine the semantic connection between words and labels in a latent space. With the help of word embedding and label embedding technique, all words and labels are represented in the \(r\)-dimensional latent space as \(H = (H^{(f)}; H^{(b)})\) and \(L\) respectively. To conveniently align the latent space of words and that of labels, a bridge mapping matrix \(W_q \in \mathbb{R}^{r \times r}\) is trained via \(Q = W_qL\).

Inspired by the interaction mechanism \([6]\), we take \(Q \in \mathbb{R}^{r \times k}\) as the attention queries for each label, and use \(H\) to construct the key-value pairs in terms of forward and backward information for each word. Then, the interactive matching score \(M^{(I)} \in \mathbb{R}^{n \times k}\)

\[
M^{(I)} = \left[\begin{array}{c}
H^{(f)}^T \\
H^{(b)}^T
\end{array}\right] \left[\begin{array}{c}
Q' \\
Q
\end{array}\right]
\]

(7)

To make sure the attention weight value fall into the range of \([0, 1]\), we normalize \(M^{(I)}\) to obtain the interaction-attention weight \(A^{(I)} = \left(A_{ij}^{(I)}\right)_{i=1\ldots n, j=1\ldots k}\) as follows.

\[
A_{ij}^{(I)} = e^{M_{ij}^{(I)}} / \sum_{i=1}^{n} e^{M_{ij}^{(I)}}
\]

(8)

Similar to self-attention mechanism, the label-aware document representation can be calculated by linear combining the label’s context words as follows.

\[
C^{(I)}_j = HA^{(I)}_j
\]

(9)

which can be taken as the representation of the input document along the \(j\)-th label. The whole matrix \(C^{(I)} \in \mathbb{R}^{2r \times k}\) is the label-aware document representation under the interaction-attention mechanism.
3.3 Attention Fusion (FA)

The above $C^{(S)}$ and $C^{(I)}$ are label-aware document representation. The former focuses on document content, while the latter prefers to the label structure. In order to take advantage of these two parts, an attention fusion strategy is designed here to adaptively extract proper information from these two components and build accurate label-aware document representation. More specifically, a fully connected layer is used to transform the input ($C^{(S)}$ and $C^{(I)}$) to weights $\alpha \in \mathbb{R}^{k \times 1}$ and $\beta \in \mathbb{R}^{k \times 1}$ via

$$\alpha = \sigma(F_1(C^{(S)})) ; \quad \beta = \sigma(F_2(C^{(I)}))$$

where $\sigma$ is sigmoid function to ensure the weights falling into (0, 1). Among them, $\alpha_j$ and $\beta_j$ indicates the importances of self-attention and interaction-attention to final representation along the $j$-th label respectively. Therefore, we add the following constraint on them

$$\alpha_j + \beta_j = 1$$

With the aid of fusion weights, we can get the final label-aware representation of input document along the $j$-th label

$$C_j = \alpha_j \times C_j^{(S)} + \beta_j \times C_j^{(I)}.$$  \hspace{1cm} (12)

The whole matrix $C \in \mathbb{R}^{2r \times k}$ is the final label-aware document representation.

3.4 Prediction Layer

Once having $C \in \mathbb{R}^{2r \times k}$, we can build the classifier via a fully connected and output layer. The final predictions are obtained by

$$\hat{y} = \sigma(W_o(f(W_f C)))$$

where $W_f \in \mathbb{R}^{r \times 2r}, W_o \in \mathbb{R}^{1 \times r}$, $f$ is the activation function ReLU, and $\sigma$ is adopted to ensure that the output value can be taken as a probability. In this case, the binary cross-entropy loss can be used as loss function which has been proved suitable for XMTC tasks [12].

$$L_{loss} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} [y_{ij} \log(\hat{y}_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}_{ij})]$$

where $N$ is the number of training documents. The ground truth $y_{ij} = 1$ if the $i$-th document belongs to the $j$-th class, otherwise $y_{ij} = 0$.

4 Experiments

In this section, we evaluate the proposed LAHA on six benchmark datasets by comparing with the state-of-the-art extreme multi-label learning methods in terms of widely used metrics. Meanwhile, the large-scale label set is divided into different groups according to their frequencies, so that LAHA and baselines can be sufficiently verified on each group.
4.1 Datasets

A series of experiments were carried out on six multi-label datasets including three small-scale datasets RCV1 [10], AAPD [24] and EUR-lex [14], and three medium-scale datasets KanShan-Cup [2], Wiki-30K [31] and Amazon-12K [13]. The raw text data for RCV1, EUR-lex, Wiki-30K and Amazon-12K are obtained from the author of XML-CNN [3]. The TF-IDF features and labels are obtained in the Extreme Classification Repository [4]. The dataset statistics are summarized in Table 1. Among them, the label information of EUR-lex and Wiki-30K is significantly sparse because they have a large set of tail labels, which make the extreme multi-label text classification more challenging.

Table 1. Summary of Experimental Datasets. $N$ is the number of training documents, $M$ is the number of testing documents, $D$ is the number of features, $L$ is the number of class labels, $\bar{L}$ is the average number of labels per document, $\hat{L}$ is the average number of documents per label, $\bar{W}$ is the average number of words per document in the training set, $\hat{W}$ is the average number of words per document in the testing set.

| datasets      | N    | M    | D    | L    | $\bar{L}$ | $\hat{L}$ | $\bar{W}$ | $\hat{W}$ |
|---------------|------|------|------|------|-----------|-----------|-----------|-----------|
| RCV1          | 23,149 | 781,265 | 47,236 | 103 | 3.2 | 729.7 | 259.5 | 269.2 |
| EUR-Lex       | 11,585 | 3,865 | 171,120 | 3,956 | 5.3 | 15.6 | 1,225.2 | 1,248.1 |
| AAPD          | 54,840 | 1,000 | 69,399 | 54 | 2.41 | 2444.0 | 163.4 | 171.7 |
| Wiki-30K      | 12,959 | 5,992 | 100,819 | 29,947 | 18.7 | 8.1 | 2246.6 | 2210.1 |
| Amazon-12K    | 490,310 | 152,981 | 135,895 | 12,277 | 5.4 | 214.5 | 230.3 | 224.7 |
| Kan-Shan Cup  | 2,799,967 | 200,000 | 411,721 | 1,999 | 2.3 | 3513.1 | 38.1 | 35.5 |

4.2 Methodology

**Baseline Algorithms** The proposed LAHA is a deep neural network model, thus the recent deep learning-based XMT methods (XML-CNN [12] and AttentionXML [25]) are selected as baselines. Note that these three methods, unlike EXAM [6], do not consider the label content information, because in some real applications, we can not get such auxiliary information, which is the main reason that EXAM is not selected as baseline. Meanwhile, the existing powerful SLEEC [2] (an embedding-based method) and PfastreXML [9] (a tree-based method) are used as baselines because they obtained the best performance in each type as shown in the Extreme Classification Repository [4]. We implement all the deep learning-based methods by pytorch following the parameter settings described in their original papers and use source codes of SLEEC and PfastreXML published by their authors.

---

2. [https://biendata.com/competition/zhihu/](https://biendata.com/competition/zhihu/)
3. [https://drive.google.com/drive/folders/1KQMBZgACUm-ZZcSrQpDPIB6CFKv9Gfb](https://drive.google.com/drive/folders/1KQMBZgACUm-ZZcSrQpDPIB6CFKv9Gfb)
4. [http://manikvarma.org/downloads/XC/XMLRepository.html](http://manikvarma.org/downloads/XC/XMLRepository.html)
Parameter Settings For all the six datasets, we preprocessed the raw text data and got 300-dimensional dense vector for each word (i.e., \( d = 300 \)) with Glove \[16\]. The number of Bi-LSTM hidden units is set to \( r = 256 \). For the self-attention mechanism, \( d_a = 256 \). For the largely sparse data EUR-Lex and Wiki-30K, attention fusion mechanism contains two FC layers with parameters \( (2r \rightarrow 256 \text{ and } 256 \rightarrow 1) \) for effective obtaining the fusion weights. For other four datasets, only one FC layer \( (2r \rightarrow 1) \) is used. In the prediction layer, ReLU is adopted as non-linear activation function. The whole deep model is trained using Adam with the initial learning rate \( (0.001) \) and the batch size \( (128) \).

Evaluation Metrics In XMTC tasks, due to the high dimension of label space and only a small subset of labels can be active in each document, a short ranked list of potential relevant labels is necessary. Thus, rank-based evaluation metrics are used to evaluate LAHA and baselines, including Precision at \( \tau \) \((P@\tau)\) and normalized Discounted Cumulative Gain at \( \tau \) \((nDCG@\tau)\). Both of them have been widely used in XMTC tasks. Among them, \( P@\tau \) is defined as

\[
P@\tau = \frac{1}{\tau} \sum_{l \in r_{\tau}(\hat{y})} y_l
\]

where \( y \in \{0, 1\}^k \) is the ground truth label vector of a document and \( r_{\tau}(\hat{y}) \) is the label indexes of top \( \tau \) highest scores of current prediction result. \( nDCG@k \) is defined as follows.

\[
nDCG@\tau = \frac{\sum_{l \in r_{\tau}(\hat{y})} y_l / \log(l + 1)}{\sum_{l=1}^{\min(\tau, \|y\|_0)} 1 / \log(l + 1)}
\]

where \( \|y\|_0 \) counts the number of relevant labels in the ground truth label vector \( y \). Larger \( P@\tau \) and \( nDCG@\tau \) indicates better performance.

4.3 Ablation Test of LAHA

In this section, we firstly demonstrate the effect of each component on LAHA. To reach this goal, we do ablation test for self-attention mechanism (SA), interaction-attention mechanism (IA) and attention fusion mechanism (FA) respectively with two datasets: one sparse data EUR-Lex and one dense data AAPD.

Fig 2 lists the prediction results on these two datasets in terms of \( P@\tau \) \((\tau = \{1, 3, 5\})\). It can be seen that SA performs well on dense multi-label data \((AAPD)\). However, neither SA nor IA can obtain good result on sparse multi-label data \((EUR-Lex)\). Fortunately, combining SA and IA improves the prediction performance (SA+IA gets better performance than SA and IA). SA prefers to extract the useful content information when constructing the label-aware document representation, but SA ignores the label structure during the learning process. IA implements this by using the label embedding learnt from the label co-exist graph. However, in real application, such graph may contain noisy information (say in dense data). Therefore, coupling with both attention
components does really helpful for final performance because they can benefit each other on different datasets.

To adaptively extract proper information to learn the final label-aware document representation, the attention fusion mechanism is introduced in LAHA. Fig. 3 lists the distribution of weights on SA and IA. It can be seen that for sparse data (EUR-Lex), the interaction-attention plays much more important role than self-attention on learning process, vice verse for dense data (AAPD). This result further clarifies that IA mechanism can leverage the label structure to improve the prediction performance for sparse data. On the other hand, in AAPD, each label has sufficient documents, i.e., SA mechanism can sufficiently capture the label-aware document information and perform well. That is why larger weights are assigned to SA on dense data. Similar trend can be found on other four datasets, which are omitted due to the page limitation.

4.4 Comparising Deep Methods on Sparse Dataset

In order to explore the effect of LAHA on sparse dataset, we further divide labels into three groups according to their occurring frequencies. The label distributions of label frequency on EUR-Lex and Wiki-30K are shown in Fig. 4.
Fig. 4. The distribution of label frequency in two sparse datasets (EUR-Lex and Wiki30K). x-axis indicates the label index sorted by occurring frequency in training data. All labels are divided into three groups (in each group, \( F \) indicates the label frequency, \( P \) is the proportion that labels in group stands at the whole label set).

Fig. 5. Comparing XML-CNN, AttentionXML and LAHA on different label groups of sparse datasets (EUR-Lex and Wiki-30K) in terms of \( P@\tau \) (\( \tau = \{1, 3, 5\} \)).

first group (\( G1 \)) contains labels occurring in less than 6 documents. The labels occurring frequently form the third group (\( G3 \)) whose total frequency is 90% of the whole data. The middle part is taken as the second group (\( G2 \)). Obviously, label prediction in \( G1 \) is much harder than in other two groups due to the lack of training documents. Obviously, Wiki-30K is much sparser than EUR-Lex.

Fig.5 shows the prediction performance obtained by three deep methods. All methods become better from \( G1 \) to \( G3 \), which is reasonable since \( G3 \) contains more training documents than \( G1 \). LAHA has an overall improvement for all groups compared with two baselines. Especially on the sparse data Wiki-30K, LAHA obtains an average of more than 39.29\%, 418.14\%, 123.80\% gain on three metrics for \( G1 \), \( G2 \), and \( G3 \) respectively to XML-CNN (the second-best method). This result further demonstrates the superiority of the proposed hybrid attention mechanism on XMTC with large-scale tail labels.
(a) The words with largest label-aware attention weights output by AttentionXML (word→{labels}) are: autism→{autism}; cure others→{disease}; disorder→{abnormal}; social communication, approach others→{social norm}; children→{children, childhood}.

(b) The words with largest label-aware attention weights output by LAHA (word→{labels}) are: autism→{autism}; disorder→{abnormal}; child life, infancy childhood, children→{children, childhood}; diagnostic, genetic factor, lack intuition→{disease}; synapses connect organize→{neurology}; asperger syndrome→{asperger}; social communication→{social norm}; security→{disease, health}.

Fig. 6. Heat map of label-aware attention weights obtained by (a) AttentionXML and (b) LAHA on an example document from Wiki30K.

To further investigate the attention-based methods, we visualize the attention weights on the original document using heat map, as shown in Fig. 6. This example document belongs to 28 labels named as autism, children, childhood, disease, asperger, social norm, health, neurology, abnormal and etc. From the attention weights, we can see that AttentionXML only captures few key words for few related labels. As expected, LAHA focuses on the related information as much as possible due to the capacity making full use of label structure and document content.

4.5 Comparison Results and Discussion

In this section, the proposed LAHA is evaluated on six benchmark datasets by comparing with four baselines in terms of \(P@\tau\) and \(nDCG@\tau\) (\(\tau = \{1, 3, 5\}\)). Table 2 shows the averaged performance of all test documents. According to the formula (15) and (16), we know \(P@1 = nDCG@1\), thus only \(nDCG@3\) and \(nDCG@5\) are listed. In each line, the best result is marked in bold, and the second best is underlined.

From Table 2, we can make a number of observations about these results. Firstly, LAHA outperforms the traditional powerful embedding-based and tree-
Table 2. Comparing LAHA with four baselines in terms of various metrics on six benchmark datasets.

| Datasets      | Metric | SLEEC | PfastreXML | XML-CNN | AttentionXML | LAHA  |
|---------------|--------|-------|------------|---------|--------------|-------|
| RCV1          | P@1   | 95.35%| 94.92%     | 95.75%  | 96.41%       | 96.95%|
|               | P@3   | 79.14%| 79.89%     | 78.63%  | 80.91%       | 81.43%|
|               | P@5   | 54.36%| 55.93%     | 54.94%  | 56.38%       | 56.44%|
|               | nDCG@3| 90.01%| 90.73%     | 89.89%  | 91.88%       | 92.69%|
|               | nDCG@5| 90.82%| 91.36%     | 90.77%  | 92.70%       | 93.01%|
| EUR-Lex       | P@1   | 75.18%| 69.03%     | 70.40%  | 67.34%       | 74.95%|
|               | P@3   | 61.67%| 57.39%     | 54.98%  | 52.52%       | 61.48%|
|               | P@5   | 50.23%| 48.69%     | 44.86%  | 42.72%       | 50.71%|
|               | nDCG@3| 63.79%| 59.51%     | 58.62%  | 56.21%       | 64.89%|
|               | nDCG@5| 58.03%| 55.72%     | 53.10%  | 50.78%       | 59.28%|
| AAPD          | P@1   | 81.96%| 82.35%     | 76.25%  | 83.02%       | 84.48%|
|               | P@3   | 57.48%| 58.01%     | 54.34%  | 58.72%       | 60.72%|
|               | P@5   | 38.99%| 40.13%     | 37.84%  | 40.56%       | 41.19%|
|               | nDCG@3| 77.65%| 78.26%     | 72.01%  | 78.01%       | 80.11%|
|               | nDCG@5| 81.59%| 82.03%     | 76.40%  | 82.31%       | 83.70%|
| Wiki-30K      | P@1   | 85.26%| 82.81%     | 82.90%  | 81.93%       | 83.18%|
|               | P@3   | 73.91%| 67.48%     | 67.46%  | 65.78%       | 72.94%|
|               | P@5   | 62.55%| 59.93%     | 57.09%  | 56.04%       | 62.77%|
|               | nDCG@3| 76.01%| 72.15%     | 71.04%  | 70.75%       | 75.44%|
|               | nDCG@5| 68.27%| 63.83%     | 62.92%  | 61.98%       | 67.87%|
| Kan-Shan Cup  | P@1   | 51.41%| 52.29%     | 49.68%  | 53.69%       | 54.38%|
|               | P@3   | 32.81%| 32.99%     | 32.27%  | 34.10%       | 34.60%|
|               | P@5   | 24.29%| 24.58%     | 24.17%  | 25.16%       | 25.88%|
|               | nDCG@3| 49.32%| 49.96%     | 46.65%  | 51.03%       | 51.70%|
|               | nDCG@5| 49.74%| 50.11%     | 49.60%  | 53.96%       | 54.65%|
| Amazon-12K    | P@1   | 93.49%| 93.95%     | 93.15%  | 93.75%       | 94.87%|
|               | P@3   | 78.01%| 78.33%     | 76.11%  | 78.36%       | 79.16%|
|               | P@5   | 62.09%| 62.77%     | 60.51%  | 62.14%       | 63.16%|
|               | nDCG@3| 86.89%| 88.41%     | 86.75%  | 87.62%       | 89.13%|
|               | nDCG@5| 84.53%| 86.23%     | 84.01%  | 86.06%       | 87.57%|
| Win times     | 6     | 0     | 0          | 0       | 0             | 24    |
based methods in most cases, while slightly underperforms the embedding-based method \textit{SLEEC} on \textit{EUR-Lex} and \textit{Wiki-30K}. From Table 1, we can see there are only 11,585 and 12,959 training documents in these two datasets, in this case, the deep model may be not sufficiently trained. Second, \textit{LAHA} is consistently superior to the state-of-the-art deep XMTC methods. The main reason is that \textit{LAHA} has ability to sufficiently determine the label-aware document representation while \textit{XML-CNN} does not. Even though \textit{AttentionXML} tries to find the relation between each pair of document and label, it only focuses on document content, which will degrade its performance on tail labels due to lack of information. Fortunately, \textit{LAHA} addresses this issue by simultaneously considering label structure via a hybrid attention mechanism.

5 Conclusions and Future Work

In this paper, a new XMTC method, \textit{LAHA}, is proposed. \textit{LAHA} utilizes self-attention and interaction-attention to extract the semantic relation between words and labels, and a hybrid-attention to construct the label-aware document representation. Extensive experiments on six benchmark datasets prove the superiority of \textit{LAHA} by comparing with the state-of-the-art XMTC methods. In a nutshell, the novelty of \textit{LAHA} lies in its providing a label-aware document representation that captures both document content and label structure, and has better discriminative ability than baselines. In real applications, more contents can be collected such as label content, which is proved to be helpful in XMTC \cite{6}. We therefore plan to extend the current model with such information.

References

1. Balasubramanian, K., Lebanon, G.: The landmark selection method for multiple output prediction. arXiv preprint arXiv:1206.6479 (2012)
2. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: Advances in neural information processing systems. pp. 730–738 (2015)
3. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
4. Cisse, M.M., Usunier, N., Artieres, T., Gallinari, P.: Robust bloom filters for large multilabel classification tasks. In: Advances in Neural Information Processing Systems. pp. 1851–1859 (2013)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Du, C., Chin, Z., Feng, F., Zhu, L., Gan, T., Nie, L.: Explicit interaction model towards text classification. arXiv preprint arXiv:1811.09386 (2018)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
8. Hsu, D.J., Kakade, S.M., Langford, J., Zhang, T.: Multi-label prediction via compressed sensing. In: Advances in neural information processing systems. pp. 772–780 (2009)
9. Jain, H., Prabhu, Y., Varma, M.: Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 935–944. ACM (2016)
10. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of machine learning research 5(Apr), 361–397 (2004)
11. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130 (2017)
12. Liu, J., Chang, W.C., Wu, Y., Yang, Y.: Deep learning for extreme multi-label text classification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 115–124. ACM (2017)
13. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 165–172. ACM (2013)
14. Mencia, E.L., Fürnkranz, J.: Efficient pairwise multilabel classification for large-scale problems in the legal domain. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 50–65. Springer (2008)
15. Munkhdalai, T., Yu, H.: Neural semantic encoders. In: Proceedings of the conference. Association for Computational Linguistics. Meeting. vol. 1, p. 397. NIH Public Access (2017)
16. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
17. Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., Varma, M.: Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 993–1002. International World Wide Web Conferences Steering Committee (2018)
18. Prabhu, Y., Varma, M.: Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 263–272. ACM (2014)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
20. Tai, F., Lin, H.T.: Multilabel classification with principal label space transformation. Neural Computation 24(9), 2508–2542 (2012)
21. Wang, L., Cao, Z., De Melo, G., Liu, Z.: Relation classification via multi-level attention cnns (2016)
22. Wang, S., Jiang, J.: Learning natural language inference with lstm. arXiv preprint arXiv:1512.08849 (2015)
23. Wang, X., Jiang, W., Luo, Z.: Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2428–2437 (2016)
24. Yang, P., Sun, X., Li, W., Ma, S., Wu, W., Wang, H.: Sgm: sequence generation model for multi-label classification. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 3915–3926 (2018)
25. You, R., Dai, S., Zhang, Z., Mamitsuka, H., Zhu, S.: Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks. arXiv preprint arXiv:1811.01727 (2018)

26. Zhang, W., Yan, J., Wang, X., Zha, H.: Deep extreme multi-label learning. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. pp. 100–107. ACM (2018)

27. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in neural information processing systems. pp. 649–657 (2015)

28. Zhang, Y., Schneider, J.: Multi-label output codes using canonical correlation analysis. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 873–882 (2011)

29. Zhou, C., Sun, C., Liu, Z., Lau, F.: A c-lstm neural network for text classification. arXiv preprint arXiv:1511.08630 (2015)

30. Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., Xu, B.: Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. arXiv preprint arXiv:1611.06639 (2016)

31. Zubiaga, A.: Enhancing navigation on wikipedia with social tags. arXiv preprint arXiv:1202.5469 (2012)