Exploiting Dynamic and Fine-grained Semantic Scope for Extreme Multi-label Text Classification

Yuan Wang[1,2], Huiling Song[1], Peng Huo[1], Tao Xu[1(✉)], Jucheng Yang[1], Yarui Chen[1], and Tingting Zhao[1]

[1] Tianjin University of Science and Technology, Tianjin 300457, China
xutaowk0@mail.tust.edu.cn
[2] Population and Precision Health Care, Ltd., Tianjin 300000, China

Abstract. Extreme multi-label text classification (XMTC) refers to the problem of tagging a given text with the most relevant subset of labels from a large label set. A majority of labels only have a few training instances due to large label dimensionality in XMTC. To solve this data sparsity issue, most existing XMTC methods take advantage of fixed label clusters obtained in early stage to balance performance on tail labels and head labels. However, such label clusters provide static and coarse-grained semantic scope for every text, which ignores distinct characteristics of different texts and has difficulties modelling accurate semantics scope for texts with tail labels. In this paper, we propose a novel framework TReaderXML for XMTC, which adopts dynamic and fine-grained semantic scope from teacher knowledge for individual text to optimize text conditional prior category semantic ranges. TReaderXML dynamically obtains teacher knowledge for each text by similar texts and hierarchical label information in training sets to release the ability of distinctly fine-grained label-oriented semantic scope. Then, TReaderXML benefits from a novel dual cooperative network that firstly learns features of a text and its corresponding label-oriented semantic scope by parallel Encoding Module and Reading Module, secondly embeds two parts by Interaction Module to regularize the text's representation by dynamic and fine-grained label-oriented semantic scope, and finally find target labels by Prediction Module. Experimental results on three XMTC benchmark datasets show that our method achieves new state-of-the-art results and especially performs well for severely imbalanced and sparse datasets.

Keywords: Extreme multi-label text classification · Semantic scope · A dual cooperative network · Data sparsity

1 Introduction

Recent years have witnessed remarkable progress in XMTC, with a variety of approaches presented in the literatures and applied in real-world scenarios, such as dynamic search advertising [21] and query recommendation [8].

W. Lu et al. (Eds.) NLPCC 2022, LNCS 13552, pp. 85–97, 2022.
https://doi.org/10.1007/978-3-031-17189-5_7

Different from classical multi-label problems, only a few are head labels with sufficient positive training data, and most labels are tail labels with few positive training data due to large label dimensionality [18,19,23] in XMTC. This data sparsity issue leads to insufficient feature learning of tail labels, and hurts prediction performance on overwhelming tail label predictions.

To solve this problem, most existing XMTC methods [3,9,9,15,18,20,21,23] take advantage of label clusters obtained in early stage to balance performance on tail labels and head labels. The main motivation is that the semantics of head labels is easy to be recognized in the semantic space due to sufficient training data, while the semantics of tail labels is vague. The precise semantics of tail labels can be learned from head labels that may appear in the same cluster. However, existing label clusters are all pre-defined global category patterns due to fixed features of labels. The static and coarse-grained semantic scope provided by such label clusters is not always consistent with dynamic real-world semantic scenarios, where content of different text has different semantic granularity. The previous model establishes structures hierarchy for the labels of a single field, and if the user is likely to be interested in overlapping topics in that and other fields, then when he enters a query into the search engine, he only gets keywords for a single domain due to the prepared label clusters. Thus, we consider developing dynamic semantic scope in the form of fine-grained teacher knowledge to improve tail label predictions accuracy and alleviate the data sparsity issue. We introduce text relevance to increase exposure of tail labels and implement a dynamic label cluster structure to personalise relevant label subsets. In detail, for given instance, We can use the relevant labels of its neighbouring text to link more rare labels. We assume that if a text is related to a label, then the text is also related to its parent label. With the help of hierarchical label information, teacher knowledge is modeled to provide dynamic and fine-grained semantic scope to rich text semantics.

In summary, We propose a novel framework TReaderXML for XMTC containing a novel dual cooperative network based on multi-head self attention mechanism to embed both guidance knowledge and text into a shared semantic space for feature interaction, effectively improving the effect of teacher knowledge. The remainder of the paper is organized as follows. In Sect. 2, we review recent related work. Section 3 introduces TReaderXML. In Sect. 4, experimental results on three XMTC benchmark datasets are shown. Section 5 concludes this work.

2 Related Work

Many methods have been proposed for addressing the data sparsity issue of XMTC. They can be categorized into the following two types: 1) flat based label clusters [3,23]; 2) tree based label clusters [9,15,17,18,20,21]. Tree based label clusters include loss function-based and structure-based.

In flat based label clusters, SLEEC [3] uses text features for clustering. A new text is projected in corresponding clusters, and labels of a new text are obtained

by K-Nearest Neighbor to alleviate the data sparsity issue. Based on SLEEC, AnnexML [23] uses label features for clustering based on graph embedding to improve the quality of clusters. In addition, in tree based label clusters, loss function-based method FastXML [21] learns an ensemble of trees which clusters the label space by optimizing a normalized Discounted Cumulative Gain (nDCG) loss function, and PfastreXML [9] replaces the nDCG loss in FastXML by its propensity scored variant which assigns higher rewards for tail label predictions. Furthermore, for structure-based methods in tree based label clusters, Parabel [15] generates a label tree by recursively clustering labels into two balanced groups to address the data sparsity issue. However, the clustering depth of Parabel is deep, which leads to error cascade problems and affects tail label predictions. Bonsai [18] uses shallow and diverse probabilistic label trees (PLTs) by removing the balance constraint in the tree construction of Parabel, which improves tail label predictions. This tree structure-based label cluster optimization is also applied to AttentionXML [17]. AttentionXML optimizes the structure of PLTs to obtain shallow and wide clusters, which improves tail label predictions.

These label cluster methods provide static and coarse-grained semantic scope for every text. It is not always consistent with dynamic real-world semantic scenarios, and reduces the precision of prior knowledge.

3 Methodology

3.1 Notation

Given a training set $\{(x_i, y_i)\}_{i=1}^{N}$ where x_i is text input sequence, and $y_i \in \{0,1\}^L$ is the label of x_i represented by L dimensional multi-hot vectors. Each dimension in y_i corresponds to a label where $y_{ij} = 1$ when the j-th label $L_{y_{ij}}$ is associated with x_i. In this paper, we introduce teacher knowledge in a training set $\{(x_i, y_i, y_i')\}_{i=1}^{N}$ where y_i' represents the text x_i's corresponding teacher knowledge.

3.2 TReaderXML

TReaderXML adopts dynamic and fine-grained semantic scope from teacher knowledge for an individual text to optimize text prior category semantic ranges. Before teacher knowledge helps read text semantics, we need a powerful feature extraction to obtain high dimensional features of semantic scope from teacher knowledge and a text respectively, and embed both of them into a shared semantic space. Then the high dimensional semantics of scope with prior knowledge helps read high dimensional semantics of a text. Based on the above motivations, we design four layers: 1) Encoding, 2) Reading, 3) Iteraction and 4) Predicting. Furthermore, a dual cooperative network contains two layers of Reading and Iteraction, and Fig. 1 shows the framework of TReaderXML.

Encoding. In this part, we design a structure of representation to obtain fine-grained semantic scope extended by teacher knowledge matrix $E_{y_i'}$ and E_{x_i}. Given a training text x_i, its vectorization is shown as follows:

$$V_{x_i} = \frac{\sum_{c=1}^{Len(x_i)} \text{Encode}(\text{x}_{ic})}{Len(\text{x}_\text{i})}. \tag{1}$$

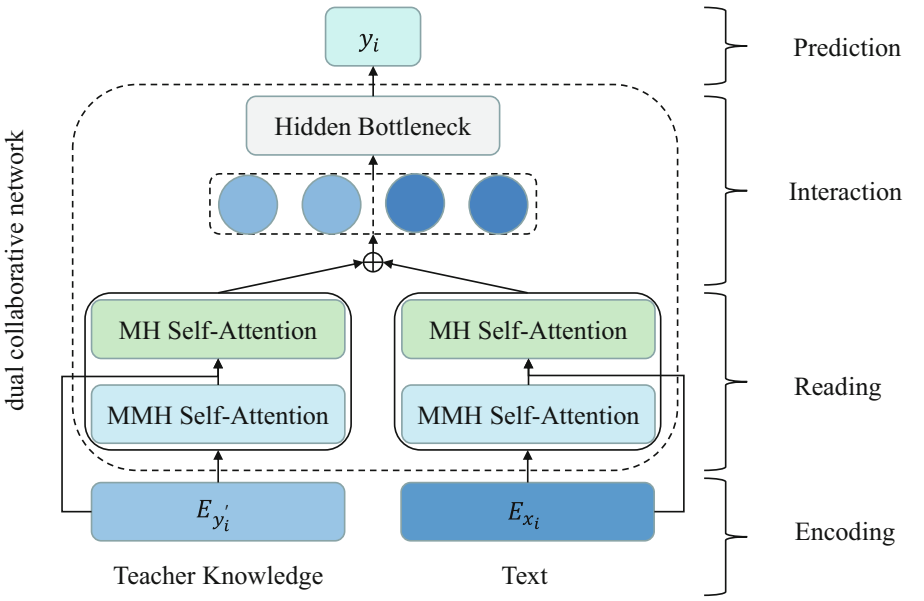

Fig. 1. An overview of TReaderXML.

Traverse x_z from training set and validation set to find the nearest neighbourhoods of x_i by using consine similarity:

$$score_{cos}(V_{x_i}, V_{x_z}) = \frac{(V_{x_i} \cdot V_{x_i})}{\| V_{x_i} \| \| V_{x_z} \|}. \tag{2}$$

and return top k nearest neighbourhoods $y_i^{nearest}$. To get the semantic scope, Gargiulo [7] uses all of ancestor labels of text labels y_i in a label tree to introduce hierarchical label information, effectively utilizing label semantic structural relation information. However, it leads to error cascade problems [18] due to the deep hierarchical structure. Furthermore, the semantics of deep hierarchical labels is often abstract, and it reduces the precision of prior knowledge. Inspired by these observations, we only use parent labels of child labels y_i in a hierarchical label tree to introduce hierarchical label information. With the advantage of low error and high precision of hierarchical label information, teacher knowledge is

modeled to provide dynamic and fine-grained semantic scope to help read text semantics. As shown in Algorithm 1, we firstly find the most relevant labels of $x_i^{nearest}$ and its non-empty parent labels. And then we put them into the label subset $SET^{nearest}$. Each label description information can be generated with widely used tricks in Parabel [15]. To keep an input sequence consistent with the semantic scope of teacher knowledge, we also initialize an embedding for an input sequence x_i, and the processing formula is shown as: $E_{x_i} = \text{Encode}(x_i)$.

Reading. In this part, we design a structure of Reading to obtain high dimensional features of semantic scope from a teacher knowledge and a text respectively and embed both of them into a shared semantic space for the preparation of feature interaction. This component in a dual cooperative network plays a key role including a mask multi-head self attention (MMHSA) layer, a multi-head self attention (MHSA) layer and residual network.

To obtain high dimensional features of semantic scope from teacher knowledge and a text respectively, we design the structure of Reading based on the self-attention mechanism [2], which contains a MMHSA layer, a MHSA layer and a residual layer. MMHSA masks the future sequence information, and depends on existing sequence information to predict the next word in a sequence. We consider MMHSA as the first layer of Reading to capture more fine-grained semantic information due to the masking in MMHSA. Furthermore, MHSA makes each word contain other semantic information of words in a text input sequence, and we consider MHSA as the second layer of Reading to capture overall semantic information. The processing formula of masking in MMHSA is shown as follows:

$$d_k = d_{\text{model}}//h, \tag{3}$$

$$Q_{y_i'} = E_{y_i'} W_{y_i'}^q,\ Q_{x_i} = E_{x_i} W_{x_i}^q, \tag{4}$$

$$K_{y_i'} = E_{y_i'} W_{y_i'}^k,\ K_{x_i} = E_{x_i} W_{x_i}^k, \tag{5}$$

$$V_{y_i'} = E_{y_i'} W_{y_i'}^v,\ V_{x_i} = E_{x_i} W_{x_i}^v, \tag{6}$$

$$Score_{y_i'} = \frac{Q_{y_i'} \cdot K_{y_i'}^T}{\sqrt{d_k}},\ Score_{x_i} = \frac{Q_{x_i} \cdot K_{x_i}^T}{\sqrt{d_k}}, \tag{7}$$

$$Score_{y_i'} = Mask(Score_{y_i'}, W_{y_i'}^{mask}), \tag{8}$$

$$Score_{x_i} = Mask(Score_{x_i}, W_{x_i}^{mask}), \tag{9}$$

$$H_i^{y_i'} = \text{Softmax}(Score_{y_i'}) \cdot V_{y_i'}, \tag{10}$$

$$H_i^{x_i} = \text{Softmax}(Score_{x_i}) \cdot V_{x_i}. \tag{11}$$

where d_{model} is dimension of embedding, and h is the number of attention heads. $W_{y_i'}^q$, $W_{y_i'}^k$, $W_{y_i'}^v$, $W_{x_i}^q$, $W_{x_i}^k$, and $W_{x_i}^v$ are weight matrices of random initialization. $W_{y_i'}^{mask}$ and $W_{x_i}^{mask}$ are upper triangular matrices. For $Mask(A, B)$, positions where the value of B is 0 are mapped into A, and the value of these positions are set to minus infinity in A. Then infinity values in A will become 0 after Softmax, and masking has been achieved. The attention output $H_i^{y_i'}$ and $H_i^{x_i}$

Algorithm 1. Encoding generation of teacher knowledge

Input: $y_i^{nearest}$
Output: $E_{y_i'}$

initialize $SET^{nearest}$={};
for $j = 0$; $j < L$; $j++$ **do**
  if $y_{ij}^{nearest} == 1$ and $Par(L_{y_{ij}}^{nearest})! = NULL$ **then**
    add $L_{y_{ij}}^{nearest}$ to $SET^{nearest}$
    add $Par(L_{y_{ij}}^{nearest})$ to $SET^{nearest}$
  end if
end for
compute $E_{y_i'} = \frac{\sum_{k=1}^{K} \text{Encode}\,(T_k^{nearest})}{K}$
return $E_{y_i'}$

learned by each head will be concatenated and transformed by multiplying a vector respectively. The output of MMHSA can be expressed by the formula given below:

$$MMHSA_{E_{y_i'}} = \tanh\left(\left\{H_1^{y_i'}, \ldots; H_h^{y_i'}\right\} \cdot W_{y_i'}^{MH}\right), \tag{12}$$

$$MMHSA_{E_{x_i}} = \tanh\left(\{H_1^{x_i}, \ldots; H_h^{x_i}\} \cdot W_{x_i}^{MH}\right). \tag{13}$$

Compared with MMHSA, the processing formula of MHSA omits formulas (6) and (7). To compensate for the loss of semantic information caused due to the masking techniques, we therefore introduce a residual network to enhance the robustness of the model and the expressiveness of the network:

$$E_{y_i'}^{residual} = E_{y_i'} + MMHSA(E_{y_i'}), \tag{14}$$

$$E_{x_i}^{residual} = E_{x_i} + MMHSA(E_{x_i}). \tag{15}$$

The design of Reading simulates the process of reading a text. Firstly a teacher and a student respectively read verbatim to understand details of texts with MMHSA, and then read comprehensively to understand themes of texts with MHSA. Furthermore, the first layer of a dual cooperative network simulates the preparation of a teacher teaching a student to read. A teacher will prepare the key points of a text and a student will preview a text to achieve the best performance of reading. The first preparation work for the process of cooperation in a dual cooperative network has been achieved, and both semantic scope and text have been embedded into a shared semantic space.

Iteraction. The high-dimensional semantic scope generated by the interaction process between the teacher knowledge and the text provides a deeper understanding of the semantics of the text.

We assume that $O_{y_i'}$ represents the output of Reading from teacher knowledge, and O_{x_i} represents the output of Reading from an input sequence. Firstly $O_{y_i'}$ and O_{x_i} are concatenated, then prior knowledge helps read text semantics with a MMHSA layer like the first layer in Reading. The processing formula is shown as follows:

$$O_{concat} = [O_{y_i'}; O_{x_i}], \tag{16}$$

$$O_{Interaction} = O_{concat} \cdot W^{HB}. \tag{17}$$

where W^{HB} represents the weight matrix of the hidden bottleneck layer. The bottleneck layer is properly constructed to significantly reduce the model size without degrading the network performance. The design of Interaction containing a MMHSA layer simulates the process of a teacher teaching a student to read word by word. The second cooperation work in a dual cooperative network has been achieved, and the semantics of semantic scope and text has been enhanced in this network for better label prediction.

Predicting. Finally, a softmax layer is applied to predict final labels. The processing formula is shown as follows:

$$Y = \text{Softmax}(O_{Interaction} \cdot W^{Output}). \tag{18}$$

where W^{Output} is the output weight matrix of fully connected layer.

Loss Function. We measure the performance with multi-label one-versus-all loss based on max entropy principle, which are widely used in classification taskes. Specifically, for a predicted score vector Y and a ground truth label vector y_i, the processing formula is shown as follows:

$$\begin{aligned} Loss_i(Y, y_i) = &- \sum_{j=1}^{L} y_{ij} \times \log\left((1 + \exp(-Y_j))^{-1}\right) \\ &+ (1 - y_{ij}) \times \log\left(\frac{\exp(-Y_j)}{1 + \exp(-Y_j)}\right). \end{aligned} \tag{19}$$

4 Experiments

4.1 Datasets and Preprocessing

Datasets. Three XMTC benchmark datasets, which have rich hierarchical information and label descriptionare, used for experiments in this paper, including AmazonCat-13K [12], EURLex[1] and RCV1 [5]. Table 1 shows the statistics of three datasets.

[1] http://manikvarma.org/downloads/XC/XMLRepository.

Table 1. Data statistics of three XMTC datasets.

Dataset	Number of train points	Number of test points	Label dimensionality	Avg. labels per point
AmazonCat-13K	1,186,239	306,782	13,330	5.04
EURLex	15449	3865	3956	5.30
RCV1	23,149	781,265	103	3.18

Preprocessing Details. For AmazonCat-13K, we truncate each input sequence after 300 words, and label description after 4 words in the same way as Parabel [15]. Word embedding in AmazonCat-13K we use comes from AttentionXML [17]. For EURLex, we truncate each input sequence after 500 words, and each label description after 4 words. Word embedding in EURLex we use also comes from datasets. For RCV1, we truncate each input sequence after 250 words, and each label description after 16 words. Pre-trained Word2Vec [22] word embedding of 400 dimensions is used in RCV1.

The results of most these baseline methods are obtained from XMTC papers [11,14,17], and we have replicated unpublished results with original papers' codes. The word embedding training of RCV1 refers to methods [16,22]. The evaluation function implementation refers to the paper [10]. The framework of model training refers to the method [6]. The experimental code on tail labels refers to AttentionXML [17]. The implementation of MHSA refers to the paper [4], and the number of attention heads h in TReaderXML is set to 4. The initial learning rate for TReaderXML training is 0.0001. After the model converges, learning rate attenuation is used to further improve scores, and Adam [13] is used for all deep learning model training. Our experimental configuration has a GPU of RTX 2080 Ti, and 128GB memory. When duplicating AnnexML [23] on EURLex dataset, it cannot be duplicated due to memory problems.

4.2 Baselines

We compare our proposed TReaderXML to the most representative XMTC methods that address data sparsity issue including AnnexML [23], PfastreXML [9], Parabel [15], FastText [1], Bonsai [18], XML-CNN [11], and AttentionXML [17]. Table 2 compares TReaderXML with baseline methods, and the results with stars are from XMTC papers [11,14,17] directly.

The proposed TReaderXML outperforms all XMTC methods for most evaluation metrics, and for a few metrics it achieves results comparable to the current approaches. Our method TReaderXML outperforms all XMTC methods, except for being slightly worse than LightXML (P@1) on AmazonCat-13K. Compared to leading extreme classifiers, TReaderXML can up to 0.16% better in P@1 metric on RURLex. For the results of RCV1, TReaderXML has a substantial improvement at P@1. We consider that the precision of TReaderXML in the first predicting position is more accurate due to effective prior knowledge and TRead-

erXML remains close to existing XMTC methods in other evaluation metrics due to the small label dimensionality of RCV1.

4.3 Evaluation Metrics

Classification accuracy is evaluated according to Precision at k (P@k), normalized Discounted Cumulative Gain at k (nDCG@k) and Propensity Scored Precision at k (PSP@k) like AttentionXML [17]. refined

4.4 Ablation Study

We conduct an ablation study as shown in Table 3 to discuss proposed novel structures of a dual cooperative network in TReaderXML. In detail, we explore the effectiveness of the teacher knowledge branch and the Reading part.

Teacher Knowledge. Config. ID 0, 1 shows the effectiveness of teacher knowledge. With dynamic and fine-grained semantic scope from teacher knowledge, Config. ID 1 has improved 5.2% over Config. ID 0 without reading part.

Reading. Config. ID 2, 3, 4, 6 shows the plausibility of Reading structure. The structure of Config. ID 2 is similar to the effect of a person only reading word by word, and it cannot comprehensively understand themes of texts. The structure of Config. ID 3 is similar to the effect of a person only reading themes of texts, and it cannot carefully understand details of texts. The structure of Config. ID 4 is similar to the effect of a person reading themes of texts firstly then reading details of texts, and it is not always consistent with human reading habits. The structure of Config. ID 6 simulates the process of human reading, reading word by word to understand details of texts and reading comprehensively to understand themes of texts. It is feasible to simulate human reading with the Reading structure. Config. ID 5, 6 shows the effectiveness of residual layer. Config. ID 6 has improved 0.44% over Config. ID 5 with residual part.

4.5 Performance on Tail Labels

To evaluate performance of TReaderXML on tail labels, we discuss experiment results of tail labels on AmazonCat-13K dataset which has the most tail labels. From Table 4, we see that TReaderXML achieves SOTA effects at PSP@5, except for being slightly worse than PfastreXML [9] at PSP@1 and PSP@3. PfastreXML replaces the nDCG loss in FastXML [21] by its propensity scored variant which is unbiased and assigns higher rewards for the tail label predictions. However, it leads to a loss in prediction accuracy.

Table 2. Performance of TReaderXML and baseline methods over three datasets (The best results are highlighted in bold).

Datasets	Methods	P@1	P@3	P@5	nDCG@3	nDCG@5
AmazonCat-13K	AnnexML*	93.54%	78.37%	63.30%	87.29%	85.10%
	Parabel*	93.03%	79.16%	64.51%	87.72%	86.00%
	Bonsai*	92.98%	79.13%	64.46%	87.68%	85.92%
	PfastreXML*	91.75%	77.97%	63.68%	86.48%	84.96%
	XML-CNN*	93.26%	77.06%	61.40%	86.20%	83.43%
	AttentionXML*	95.92%	82.41%	67.31%	91.17%	89.48%
	X-Transformer*	96.70%	83.85%	68.58%	–	–
	APLC-XLNet	94.56%	79.82%	64.61%	88.74%	86.66%
	LigntXML	**96.77%**	84.02%	68.70%	–	–
	TReaderXML	96.64%	**85.57%**	**68.98%**	**93.99%**	**91.67%**
EURLex	AnnexML*	79.66%	64.94%	53.52%	68.70%	62.71%
	Parabel*	82.12%	68.91%	57.89%	72.33%	66.95%
	Bonsai*	82.30%	69.55%	58.35%	72.97%	67.48%
	PfastreXML*	73.13%	60.16%	50.54%	63.51%	58.71%
	XML-CNN*	68.01%	54.03%	43.93%	57.44%	51.83%
	AttentionXML*	87.12%	73.99%	61.92%	77.44%	71.53%
	X-Transformer*	87.22%	75.12%	62.90%	-	–
	APLC-XLNet	87.72%	74.56%	62.28%	77.90%	71.75%
	LigntXML	87.63%	75.89%	63.36%	-	–
	TReaderXML	**87.88%**	**78.07%**	**64.05%**	**80.70%**	**73.56%**
RCV1	AnnexML*	90.89%	76.48%	52.77%	–	–
	Parabel*	87.79%	64.84%	45.60%	77.01%	77.92%
	Bonsai*	85.23%	65.12%	45.89%	76.55%	77.59%
	PfastreXML*	68.82%	60.76%	43.28%	69.40%	71.24%
	XML-CNN*	93.63%	73.90%	52.16%	85.24	86.69%
	AttentionXML*	96.41%	**80.91%**	**56.38%**	**91.88%**	**92.70%**
	X-Transformer*	–	–	–	–	–
	APLC-XLNet	59.46%	43.79%	33.44%	–	–
	LigntXML	95.31%	78.40%	54.93%	–	–
	TReaderXML	**97.50%**	78.74%	54.67%	90.29%	90.94%

Table 3. Ablation study of TReaderXML on AmazonCat-13K (The best results are highlighted in bold).

Config. ID	Teacher knowledge	Reading	P@1	P@3	P@5	nDCG@3	nDCG@5
0	–	–	88.73%	69.12%	53.94%	78.78%	75.42%
1	True	–	93.93%	77.70%	58.53%	87.15%	81.79%
2	True	MMHSA+R	95.49%	83.45%	66.08%	92.00%	88.81%
3	True	MHSA+R	95.52%	83.46%	66.04%	92.03%	88.81%
4	True	MHSA+R+MMHSA	96.49%	85.44%	68.84%	93.86%	91.52%
5	True	MMHSA+MHSA	96.20%	84.89%	68.04%	93.33%	90.72%
6	True	MMHSA+R+MHSA	**96.64%**	**85.57%**	**68.98%**	**93.99%**	**91.67%**

Table 4. Performance on tail labels in AmazonCat-13K (The best results are highlighted in bold).

Methods	PSP@1	PSP@3	PSP@5
AnnexML*	49.04%	61.13%	69.64%
Parabel*	50.93%	64.00%	72.08%
Bonsai*	51.30%	64.60%	72.48%
PfastreXML*	**69.52%**	**73.22%**	75.48%
XML-CNN*	52.42%	62.83%	67.10%
AttentionXML*	53.76%	68.72%	76.38%
X-Transformer*	–	–	–
APLC-XLNet	52.22%	65.08%	71.40%
LigntXML	**-**	–	–
TReaderXML	57.15%	71.64%	**77.27%**

5 Conclusions

In this work, our method TReaderXML define semantic scope from teacher knowledge, which inherits the strength of hierarchical label information and meanwhile improves dynamic high level category information as semantic supplements and constraints. The proposed dual cooperative network learned semantic information in the way of people reading. Moreover, teacher knowledge can flexibly incorporate prior label information like semantic structures or descriptions.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No. 61976156 and No. 61702367), Tianjin Science and Technology Commissioner project (No. 20YDTPJC00560), the Natural Science Foundation of Tianjin (No. 19JCYBJC15300).

References

1. Armand, J., Edouard, G., Piotr, B., Matthijs, D., Herve, J., Tomas, M.: Fasttext. zip: compressing text classification models. arXiv preprint arXiv:1612.03651 (2016)
2. Ashish, V.: Attention is all you need. Adv. Neural. Inf. Process. Syst. **30**, 5998–6008 (2017)
3. 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)
4. Biqing, Z., Heng, Y., Ruyang, X., Wu, Z., Xuli, H.: Lcf: a local context focus mechanism for aspect-based sentiment classification. Appli. Sci. **9**, 3389 (2019)
5. Lewis, D.D., Yiming, Y., Rose, T.G., Fan, L.: Rcv1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)

6. 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)
7. Francesco, G., Stefano, S., Mario, C., Giuseppe, D.P.: Deep neural network for hierarchical extreme multi-label text classification. Appli. Soft Comput. **79**, 125–138 (2019)
8. Himanshu, J., Venkatesh, B., Bhanu, C., Manik, V.: Slice: scalable linear extreme classifiers trained on 100 million labels for related searches. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 528–536 (2019)
9. Himanshu, J., Yashoteja, P., Manik, V.: Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In: Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, pp. 935–944 (2016)
10. Huang, X., Chen, B., Xiao, L., Jing, L.: Label-aware document representation via hybrid attention for extreme multi-label text classification. arXiv preprint arXiv:1905.10070 (2019)
11. Jingzhou, L., Wei-Cheng, C., Yuexin, W., Yiming, 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 (2017)
12. Bhatia, K., et al.: The extreme classification repository: multi-label datasets and code (2016). http://manikvarma.org/downloads/XC/XMLRepository.html
13. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)
14. Lin, X., Xin, H., Boli, C., Liping, J.: Label-specific document representation for multi-label text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 466–475 (2019)
15. 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, pp. 993–1002 (2018)
16. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50 (2010)
17. Ronghui, Y., Zihan, Z., Ziye, W., Suyang, D., Hiroshi, M., Shanfeng, Z.: Attentionxml: label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In: Advances in Neural Information Processing Systems, pp. 5820–5830 (2019)
18. Khandagale, S., Xiao, H., Babbar, R.: Bonsai: diverse and shallow trees for extreme multi-label classification. Mach. Learn. **109**(11), 2099–2119 (2020). https://doi.org/10.1007/s10994-020-05888-2
19. Wei-Cheng, C., Hsiang-Fu, Y., Kai, Z., Yiming, Y., Dhillon, I.: Taming pretrained transformers for extreme multi-label text classification. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3163–3171 (2020)
20. Wissam, S.: Craftml, an efficient clustering-based random forest for extreme multi-label learning. In: International Conference on Machine Learning (2018)
21. Yashoteja, P., Manik, V.: 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 (2014)

22. Yoav, G., Omer, L.: word2vec explained: deriving mikolov et al'.s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 (2014)
23. Yukihiro, T.: Annexml: approximate nearest neighbor search for extreme multi-label classification. In: The 23rd ACM SIGKDD International Conference, pp. 455–464 (2017)