Reinforcement Learning for Extreme Multi-label Text Classification

Hui Teng[1,2(✉)], Yulei Li[1,2], Fei Long[1,2,3], Meixia Xu[1,2], and Qiang Ling[4]

[1] Chinaso Inc., Beijing 100077, China
tenghui@chinaso.com
[2] State Key Laboratory of Media Convergence Production Technology and Systems, Xinhua News Agency, Beijing, China
[3] Department of Computer Science, Northeast University, Shenyang 110819, China
[4] Department of Automation, University of Science and Technology of China, Hefei 230027, China

Abstract. Extreme multi-label text classification (XMC) is an important yet challenging problem in the NLP community, which refers to the problem of assigning to each document its most relevant subset of class labels from an extremely large label collection. For example, the input text could be a story document on chinastory.cn and the labels could be story categories that implies the potential meaning. However, naively applying normal neural network models to the XMC problem leads to sub-optimal performance due to the large output space and the label sparsity issue. In this paper, we presents the first attempt at applying reinforcement learning to XMC. Experimental results on public and our own engineering datasets demonstrate that our approach achieves expecting performance compared with the evaluation of the state-of-the-art methods.

Keywords: Extreme multi-label text classification · Reinforcement learning

1 Introduction

There has been a lot of interests in extreme multi-label text classification (XMC) problem: given an input text instance, return the most relevant labels from an enormous label collection, where the number of labels could be in the millions or more, which becomes increasingly important due to the fast growing of internet contents and the urgent needs for organizational views of big data. Multi-label classification is fundamentally different from the traditional binary classification problems which have been intensively studied in the machine learning literature. Binary classifiers treat class labels as independent target variables, which is clearly sub-optimal for multi-label classification as the dependencies among class

H. Teng and Y. Li—Contributed equally.

F. Sun et al. (Eds.): ICCSIP 2020, CCIS 1397, pp. 243–250, 2021.
https://doi.org/10.1007/978-981-16-2336-3_22

labels cannot be leveraged. Of the many related tasks, discovering relevant labels from document is of considerable practical importance.

In Ref. [1], XML-CNN is first adopted for XMC task which modifies the traditional TextCNN [2] architecture by using a dynamic max pooling scheme and adding a hidden bottleneck layer that captures richer information from different regions of the document. However, such a method admits the following disadvantages: (1) CNN model is unable to discover the dependency patterns due to the sparsity issue of the XMC datasets which typically exhibit a power-law distribution of labels, and the substantial proportion of the labels have very few training instances associated with them. (2) The computational costs in both training and testing of mutually independent classifiers would be practically prohibiting when the number of labels reaches hundreds of thousands or even millions.

In Ref. [5,6], interactive methods are proposed for topic discovery in order to incorporate with user's intention. However such interactive algorithms are for clustering, which introduces the uncertainty and randomness.

Specifically, representation learning is a fundamental problem in natural language processing and how to learn a structured representation for text classification is still challengeable. Unlike most existing representation models that either use no structure or rely on pre-specified structures, the reinforcement learning (RL) method that we apply is able to learn sentence representation by discovering optimized structures automatically [10]. However, there are few attempts to apply RL method into XMC tasks due to the intractable issues of XMC datasets.

The main contributions are summarized as follows:

1. We apply a reinforcement learning method and make a optimization for extreme multi-label text classification.
2. We re-examine the state of the art of XMC by conducting a comparative experimental evaluation of 5 methods which are most representative in text classification.
3. We develop a practical CMS online system for editors and perform extensive experimental validations for the proposed method.

The rest of this paper is organized as follows: Sect. 2 gives the related works and background of XMC. In Sect. 3 we give a detailed introduction about the RL method and Sect. 4 presents our extensive experiments and results, followed by conclusion and future work in Sect. 5.

2 Related Works and Background

2.1 Methods for Text Classification

Methods for comparison are outlined, including the most representative methods in XMC and some successful deep learning methods which are designed for mutliclass text classification but also applicable to XMC with minor adaptations.

a. FastText. FastText [7] is a simple yet effective baseline method for text classification, which is inspired by the recent work on efficient word representation learning, such as skip-gram and CBOW [8]. The representation of text is constructed by averaging the embeddings of the words, upon which a softmax layer is applied to map the document representation to class labels. This simplicity makes FastText very efficient to train and achieves state-of-the-art performances on both precision and time consuming. However, simply averaging input word embeddings with the shallow architecture for document-to-label mapping might limits its success in XMC. In XMC task, document presentations need to capture more high dimensional information for predicting multiple correlated labels and discriminating them from enormous numbers of irrelevant labels.

b. FastXML. FastXML [9] aims to develop an extreme multi-label classifier that is faster to train and more accurate at prediction and is considered as the state-of-the-art tree-based method for XMC. It learns a hierarchy of training instances and optimizes an NDCG-based objective at each node of the hierarchy. Specifically, a hyperplane parameterized by $w \in \mathbb{R}^D$ is induced at each node, which splits the set of documents in the current node into two subsets; the ranking of the labels in each of the two subsets are jointly learned. The key idea is to have the documents in each subset sharing similar label distribution, and to characterize the distribution using a set-specific ranked list of labels. This is achieved by jointly maximizing NDCG scores of the ranked label lists in the two sibling subsets. In practice, an ensemble of multiple induced trees are learned to improve the robustness of predictions. At prediction time, each test document is passed from the root to a leaf node in each induced tree, and the label distributions in all the reached leaves are aggregated for the test document.

c. TextCNN. TextCNN [2] applys convolutional neural networks to text classification for the first time. TextCNN depicts three filter region sizes in the convolutional layer, each of which has two filters. Filters perform convolutions on the sentence matrix and generate feature maps which are fed to a max-pooling layer. By concatenating the outputs that generated from all six maps, a feature vector is generated. The final fully-connected layer with L softmax outputs corresponding to L labels uses it as inputs to classify the sentence representation. TextCNN obtains excellent performance in text classification, and is considered as a strong baseline comparative method. Considering the two distinct type of words embedding inputs, we distinguish TextCNN into two categories: Bow-CNN [3] and XML-CNN [1], which take bag of words feature and word vector as input respectively.

d. SLEEC. SLEEC [4] is most representative for target-embedding methods in XMC which extends embedding methods in multiple ways and uses KNN for classification stage.

2.2 Chinaso Application

The database "China Story" [12] is built by Chinaso of Xinhua News Agency with the aim of "telling China's stories well, and making the voice of China heard". A challenging problem at the database amounts to discovering relevant labels from an enormous output space of potential candidates for one story document: for example, suggesting keywords to editors labeling new stories, as well as to internet users starting new campaign on Chinaso website. Normally, this task was accomplished manually, which consumes a large amount of manpower and time.

3 Reinforcement Learning

In the RL [10] method, three components are interleaved together. The policy network adopts a stochastic policy and uses a delayed reward from the final classification network to guide the policy learning for structure discovery. While the state representation of policy network is derived from the representation models which contains two models: Information Distilled LSTM which selects important, task-relevant words to build sentence representation, and Hierarchical Structured LSTM which discovers phrase structures and builds sentence representation with a two-level LSTM. The final classification network makes prediction on top of structured sentence representation and facilitates reward computation for the policy network. Figure 1 gives the detailed illustration of the overall framework.

Generally, the most straightforward adaptation from the multi-class classification problems to multi-label ones would be to extend the traditional cross-entropy loss as follows:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L} y_{ij}\log\left(\hat{p}_{ij}\right) \tag{1}$$

where Θ denotes classification model parameters, $\hat{p}_{ij}$ is the model prediction for instance i on label j via a softmax activation. Specifically, in our RL method, we modify the loss function and introduce binary cross-entropy objective as a substitute which can be formulated as follows:

$$\min_{\Theta} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{L}\left[y_{ij}\log\left(\sigma\left(f_{ij}\right)\right) + \left(1-y_{ij}\right)\log\left(1-\sigma\left(f_{ij}\right)\right)\right] \tag{2}$$

where σ is the sigmoid function $\sigma\left(x\right) = {}^{1}\!/_{1+e^{-x}}$.

4 Performance Evaluation

4.1 Datasets

We first use the well-known RCV1 [11] datasets to evaluate the above method, which contains manually categorized newswire stories made available by Reuters, Ltd.

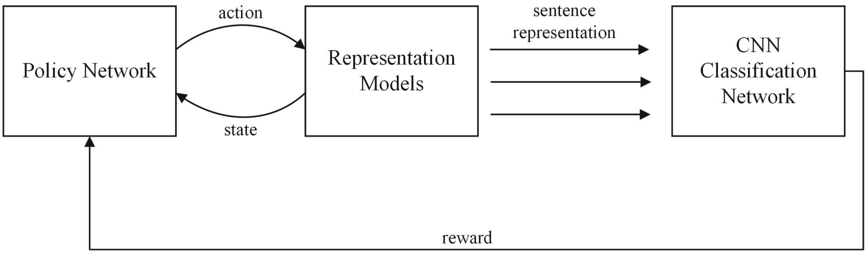

Fig. 1. Overview of the overall RL network. The policy network (PNet) samples an action at each state. The structured representation model offers state representation to PNet and outputs the final sentence representation to the classification network (CNet). CNet performs text classification and provides reward to PNet.

Table 1. Datasets statistics of RCV1

Datasets	Train instances	Test instances	Labels	Labels/document	Documents/label
RCV1	23149	781265	103	3.18	729.67
ChinaStory	8668	963	118	6.19	81.62

In addition, based on the consideration of actual needs for online application and demand for evaluating the practical performance of XMC methods, we build our own practical Chinese datasets as well. The statistical details of the two datasets are displayed in Table 1.

4.2 Experimental Validation Results

a. Evaluation Metrics. To test the performance of different methods in XMC, we introduce several evaluation metrics. In XMC datasets, each instance only has very few relevant labels which means that how to present a short ranked list of potentially relevant labels for each test instance and evaluate the quality of such ranked lists with an emphasis on the relevance of the top portion of each list is far more important. As a result rank-based evaluation metrics have been commonly used for comparing XMC methods, including the precision at top K (P@K) and the Normalized Discounted Cummulated Gains (NDCG@K) [9]. We follow such convention and use these two metrics in our evaluation in this paper, with $k = 1, 3, 5$. Denoting by $\mathbf{y} \in \{0,1\}^L$ as the vector of true labels of an document, and $\hat{\mathbf{y}} \in \mathbb{R}^L$ as the system-predicted score vector for the same document, the metrics are defined as:

$$\text{P@}k = \frac{1}{k} \sum_{l \in r_k(\hat{\mathbf{y}})} y_l \tag{3}$$

$$\mathrm{DCG@}k = \sum_{l \in r_k(\hat{\mathbf{y}})} \frac{y_l}{\log(l+1)} \tag{4}$$

$$\mathrm{NDCG@}k = \frac{\mathrm{DCG@}k}{\sum_{l=1}^{\min\left(k, \|\mathbf{y}\|_0\right)} \frac{1}{\log(l+1)}} \tag{5}$$

where $r_k(\hat{\mathbf{y}})$ is the set of rank indices of the truly relevant labels among the top-k portion of the system-predicted ranked list for a document, and $\|\mathbf{y}\|_0$ counts the number of relevant labels in the ground truth label vector $\mathbf{y}$. P@K and NDCG@K are calculated for each test document and then averaged over all the documents.

b. Evaluation Results. Using these specific evaluation metrics, we conduct extensive experiments on the two datasets. Table 2 and 3 demonstrates the P@K and G@K results of the methods on RCV1 respectively, showing that RL method achieved competitive performance which consistently produced the best or the second best results on the datasets no matter when $k = 1, 3$ or 5. Note that the RCV1 datasets have a higher number of training instances per class and the RL method is capable to learn accurate sentence representation, especially when $k = 5$. Meanwhile, based on the actual needs for online application Fig. 2 displays the results on our Chinese datasets compared with other 5 state-of-the-art method. Note that owing to the small scale of practical datasets, all the methods obtained close performance, which strongly indicates that our RL method is able to discover the representation in small datasets and can be easily scaled to the larger datasets. However, due to the very limit manual scale of China Story database, all data-driven representation learning methods are hardly capable to capture the complete feature, while FastText works better under this circumstance on account of its simplify.

Table 2. Evaluation results in P@K on datasets RCV1

Metrics	FastXML	SLEEC	FastText	Bow-CNN	XML-CNN	Ours
P@1	94.62	95.35	95.40	96.40	**96.86**	**96.54**
P@3	78.40	79.51	79.96	**81.17**	81.11	**81.13**
P@5	54.82	55.06	55.64	56.74	56.07	**56.89**

Table 3. Evaluation results in G@K on datasets RCV1

Metrics	FastXML	SLEEC	FastText	Bow-CNN	XML-CNN	Ours
G@1	94.62	95.35	95.40	96.40	**96.86**	**96.54**
G@3	89.21	90.45	90.95	92.04	**92.22**	91.89
G@5	90.27	90.97	91.68	92.89	92.63	**92.96**

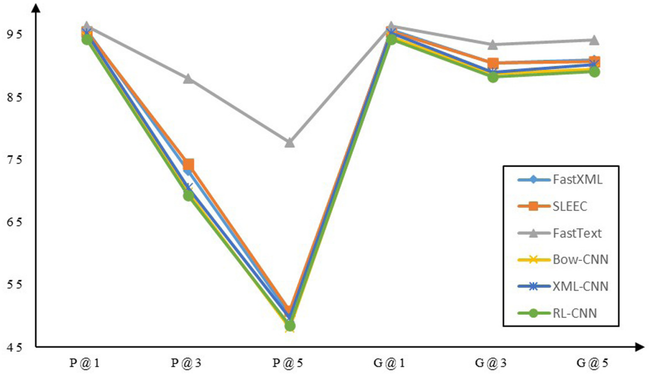

Fig. 2. Evaluation results on our ChinaStory datasets

5 Conclusion and Future Work

In this paper, we present reinforcement learning approach to discover the structured document representation for extreme multi-label text classification. Experimental validation in comparison with other state-of-the-art methods on datasets is conducted, demonstrating that on both public and practical datasets, expecting results have been obtained. However, There still exists a lot work to be further investigated. Firstly, we wish to discover more relevant label patterns from document for the end editors; Secondly, we hope to develop a more flexible graphic user interface (GUI) and integrate more high-level knowledge of the human's intention as a feedback information into the model. Finally, we wish to discover more hierarchical structure of the label in the document in a coarse-to-fine manner.

References

1. Liu, J., Chang, W. C., Wu, Y., Yang, Y.: Deep learning for extreme multi-label text classification. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 115–124 (2017)
2. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
3. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058 (2014)
4. 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)
5. Choo, J., Lee, C., Reddy, C.K., Park, H.: UTOPIAN: user-driven topic modeling based on interactive nonnegative matrix factorization. IEEE Trans. Vis. Comput. Graph. **19**(12), 1992–2001 (2013)

6. Teng, H., Liu, H., Yu, L., Sun, F.: Representative video action discovery using interactive non-negative matrix factorization. In: Hu, X., Xia, Y., Zhang, Y., Zhao, D. (eds.) ISNN 2015. LNCS, vol. 9377, pp. 205–212. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25393-0_23
7. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. 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 (2014)
10. Zhang, T., Huang, M., Zhao, L.: Learning structured representation for text classification via reinforcement learning. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
11. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
12. Chinastory. https://www.chinastory.cn/english/index.html