Joint Binary Neural Network for Multi-label Learning with Applications to Emotion Classification

Huihui He and Rui Xia(✉)

School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
hehuihui1994@gmail.com, rxia@njust.edu.cn

Abstract. Recently the deep learning techniques have achieved success in multi-label classification due to its automatic representation learning ability and the end-to-end learning framework. Existing deep neural networks in multi-label classification can be divided into two kinds: binary relevance neural network (BRNN) and threshold dependent neural network (TDNN). However, the former needs to train a set of isolate binary networks which ignore dependencies between labels and have heavy computational load, while the latter needs an additional threshold function mechanism to transform the multi-class probabilities to multi-label outputs. In this paper, we propose a joint binary neural network (JBNN), to address these shortcomings. In JBNN, the representation of the text is fed to a set of logistic functions instead of a softmax function, and the multiple binary classifications are carried out synchronously in one neural network framework. Moreover, the relations between labels are captured via training on a joint binary cross entropy (JBCE) loss. To better meet multi-label emotion classification, we further proposed to incorporate the prior label relations into the JBCE loss. The experimental results on the benchmark dataset show that our model performs significantly better than the state-of-the-art multi-label emotion classification methods, in both classification performance and computational efficiency.

Keywords: Sentiment analysis · Emotion classification
Multi-label classification

1 Introduction

Multi-label emotion classification, is a sub-task of the text emotion classification, which aims at identifying the coexisting emotions (such as joy, anger and anxiety, etc.) expressed in the text, has gained much attention due to its wide potential applications. Taking the following sentence

Example 1: "*Feeling the warm of her hand and the attachment she hold to me, I couldn't afford to move even a little, fearing I may lost her hand*"

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 250–259, 2018.
https://doi.org/10.1007/978-3-319-99495-6_21

for instance, the co-existing emotions expressed in it contain *joy*, *love*, and *anxiety*.

Traditional multi-label emotion classification methods normally utilize a two-step strategy, which first requires to develop a set of hand-crafted expert features (such as bag-of-words, linguistic features, emotion lexicons, etc.), and then makes use of multi-label learning algorithms [5,14,16,17,22] for multi-label classification. However, the work of feature engineering is labor-intensive and time-consuming, and the system performance highly depends on the quality of the manually designed feature set. In recent years, deep neural networks are of growing attention due to their capacity of automatically learn the internal representations of the raw data and integrating feature representation learning and classification into one end-to-end framework.

Existing deep learning methods in multi-label classification can be roughly divided into two categories:

- Binary relevance neural network (BRNN), which constructs an independent binary neural network for each label, where multi-label classification is considered as a set of isolate binary classification tasks and the prediction of the label set is composed of independent predictions for individual labels.
- Threshold dependent neural network (TDNN), which normally constructs one neural network to yield the probabilities for all labels via a softmax function, where the probabilities sum up to one. Then, an additional threshold mechanism (e.g., the calibrated label ranking algorithm) is further needed to transform the multi-class probabilities to multi-label outputs.

The structures of BRNN and TDNN are shown in Fig. 1(a) and (b), respectively.

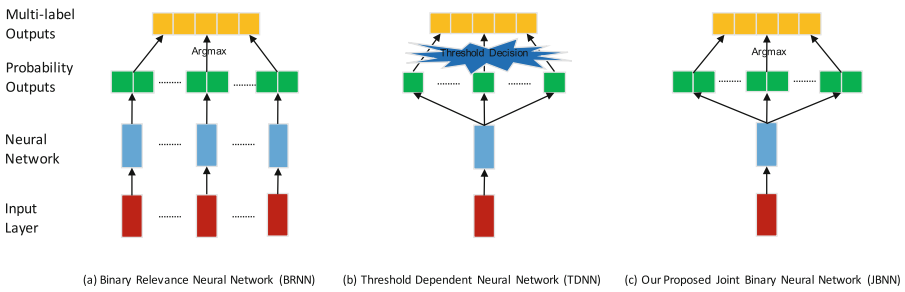

Fig. 1. Different ways of constructing neural networks for multi-label classification.

However, both kinds of methods have their shortcomings. The former one, BRNN, usually known in the literature as binary relevance (BR) transformation [12], not only ignores dependencies between labels, but also consumes much more resources due to the need of training a unique classifier and make prediction for each label. The latter one, TDNN, although has only one neural network, can only yield the category probabilities of all class labels. Instead, it needs an

additional threshold function mechanism to transform the category probabilities to multi-label outputs. However, building an effective threshold function is also full of challenges for multi-label learning [4,7,10,15,20].

In this paper, we propose a simple joint binary neural network (JBNN), to address these two problems. We display the structure of JBNN in Fig. 1(c). As can be seen, in JBNN, the bottom layers of the network are similar to that in TNDD. Specifically, we employ a Bidirectional Long Short-Term Memory (Bi-LSTM) structure to model the sentence. The attention mechanism is also constructed to get the sentence representation. After that, instead of a softmax function used in TDNN, we feed the representation of a sentence to multiple logistic functions to yield a set of binary probabilities. That is, for each input sentence, we conduct multiple binary classifications synchronously in one neural network framework. Different from BRNN, the word embedding, LSTMs, and the sentence representation are shared among the multiple classification components in the network. Moreover, the relations between labels are captured based on a joint binary learning loss. Finally, we convert the multi-variate Bernoulli distributions into multi-label outputs, the same as BRNN. The JBNN model is trained based on a joint binary cross entropy (JBCE) loss. To better meet the multi-label emotion classification task, we further proposed to incorporate the prior label relations into the JBCE loss. We evaluate our JBNN model on the widely-used multi-label emotion classification dataset Ren-CECps [9]. We compare our model with both traditional methods and neural networks. The experimental results show that:

- Our JBNN model performs much better than the state-of-the-art traditional multi-label emotion classification methods proposed in recent years;
- In comparison with the BRNN and TDNN systems, our JBNN model also shows the priority, in both classification performance and computational efficiency.

2 Model

2.1 Joint Binary Neural Network

A Bi-LSTM structure is first employed to model the sentence. On the basis of Bi-LSTM, we propose our Joint Binary Neural Network (JBNN) for multi-label emotion classification. The structure of JBNN is shown in Fig. 2.

Before going into the details of JBNN, we first introduce some notations. Suppose $E = \{e_1, e_2, \ldots, e_m\}$ is a finite domain of possible emotion labels. Formally, multi-label emotion classification may be defined as follows: giving the dataset $D = \{(x^{(k)}, y^{(k)}) \mid k = 1, \ldots, N\}$ where N is the number of examples in the D. Each example is associated with a subset of E and this subset is described as an m-dimensional vector $y^{(k)} = \{y_1, y_2, \ldots, y_m\}$ where $y_j^{(k)} = 1$ only if sentence $x^{(k)}$ has emotion label e_j, and $y_j^{(k)} = 0$ otherwise. Given D, the goal is to learn a multi-label classifier that predicts the label vector for a given example. An example is a sentence in emotion classification.

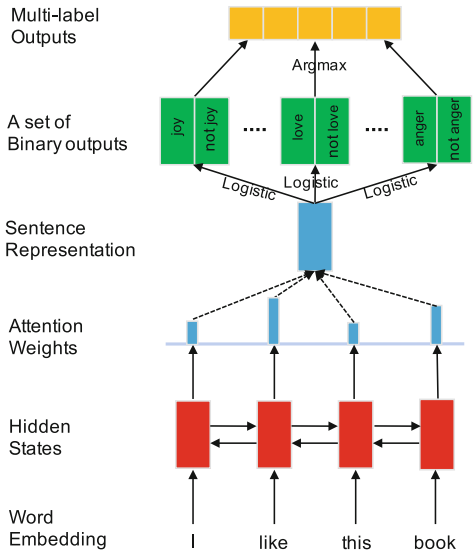

Fig. 2. Overview of the joint binary neural network.

As shown in Fig. 2, in JBNN, each word is represented as a low dimensional, continuous and real-valued vector, also known as word embedding [2,6]. All the word vectors are stacked in a word embedding matrix $L_w \in R^{d \times |V|}$, where d is the dimension of word vector and $|V|$ is vocabulary size. After we feed word embedding to Bi-LSTM, we can get hidden states $[h_1, h_2, \ldots, h_n]$ for a sentence as the initial representation.

Since not all words contribute equally to the representation of the sentence, we adopt the attention mechanism [1,18] to extract such words that are important to the meaning of the sentence. Assume h_t is the hidden states outputted in Bi-LSTM. We use an attention function to aggregate the initial representation of the words to form the attention vector v, also called sentence representation. Firstly, we use

$$u_t = \tanh\left(wh_t + b\right), \tag{1}$$

as a score function to calculate the importance of h_t in the sentence, where w and b are weight matrix and bias respectively. Then we get a normalized importance weight α_t for the sentence through a softmax function:

$$\alpha_t = \frac{\exp(u_t^T u_1)}{\sum_t \exp(u_t^T u_1)}. \tag{2}$$

After computing the word attention weights, we can get the final representation v for the sentence using equation:

$$v = \sum_t \alpha_t h_t. \tag{3}$$

After getting the sentence representation v, traditional Bi-LSTM based classification model normally feed v into a softmax function to yield multi-class probabilities for multi-class classification. Our JBNN model differs from the standard model in that, we feed the feature vector v to C logistic functions, instead of a softmax function, to predict a set of binary probabilities $\{p(y_j = 1 \mid x), j = 1, \ldots, C\}$.

$$p(y_j = 1 \mid x) = p_j = \frac{1}{1 + e^{w_j v + b_j}}, \tag{4}$$

$$p(y_j = 0 \mid x) = 1 - p_j, \tag{5}$$

where w_j and b_j are the parameters in j-th logistic component.

Each component will receive a binary probabilities which determines whether this label is True or False in the current instance (*i.e.*, whether the label belongs to the instance):

$$\hat{y_j} = \arg\max_{y_j} p(y_j \mid x). \tag{6}$$

At last, we concatenate $\hat{y_j}$ to form the final predictions $\hat{y} = [\hat{y_1}, \ldots, \hat{y_C}]$.

2.2 Joint Binary Cross Entropy Loss with Label Relation Prior

The JBNN model can be trained in a supervised manner by minimizing the following Joint Binary Cross Entropy (JBCE) loss function:

$$L = -\sum_{j}^{C} \Big(y_j \log p_j + (1 - y_j) \log(1 - p_j) \Big) + \lambda ||\theta||^2, \tag{7}$$

where λ is the weight for L_2-regularization, and θ denotes the set of all parameters. Note that different from the standard cross entropy loss defined in a multi-class classification task, our JBCE loss is defined in a set of binary classification tasks.

To better meet the multi-label emotion classification task, inspired by [22], we further proposed to incorporate the prior label relations defined in the Plutchik's wheel of emotions [8] into the JBCE loss.

Plutchik's psychoevolutionary theory of emotion is one of the most influential classification approaches for general emotional responses. He considered there to be eight primary emotions: anger, fear, sadness, disgust, surprise, anticipation, trust, and joy. The wheel Plutchik's is used to illustrate different emotions in a compelling and nuanced way. It includes several typical emotions and its eight sectors indicate eight primary emotion dimensions arranged as four pairs of opposites.

In the emotion wheel, emotions sat at opposite end have an opposite relationship, while emotions next to each other are more closely related. As shown in Fig. 3, we followed [22] by measuring the relations $w_{s,t}$ between the s-th and t-th emotions based on the angle between them.

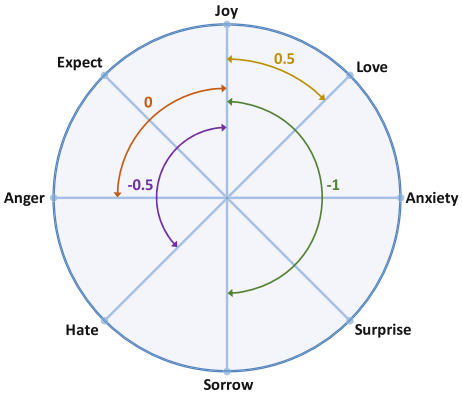

Fig. 3. Plutchik's wheel of emotions.

- In case of emotion pairs with 180° (*i.e.*, opposite to each other), define $w_{s,t} = -1$;
- In case of emotion pairs with 90°, define $w_{s,t} = 0$;
- In case of emotion pairs with 45°, define $w_{s,t} = 0.5$;
- In case of emotion pairs with 135°, define $w_{s,t} = -0.5$.

On this basis, the union loss function is defined as:

$$L = -\sum_{j=1}^{C} \Big(y_j \log p_j + (1 - y_j) \log(1 - p_j) \Big) + \lambda_1 \sum_{s,t} w_{s,t} (p_s - p_t)^2 + \lambda_2 ||\theta||^2. \tag{8}$$

The behind motivation is that if two emotions (such as joy and love) have a high positive correlation, we hope the prediction on the two emotions remain similar. On the contrary, if two emotions (such as joy and sorrow) have a high negative correlation, we hope the predictions on the two emotions remain different.

3 Experiments

3.1 Experimental Settings

We conduct the experiments on the Ren-CECps corpus [9] which was widely used in multi-label emotion classification. It contains 35,096 sentences selected from Chinese blogs. Each sentence is annotated with 8 basic emotions, such as *anger*, *anxiety*, *expect*, *hate*, *joy*, *love*, *sorrow* and *surprise*.

Due to the inherent differences in classification problems, common metrics for multi-label classification are different from those used in single-label classification. In this study, five popular evaluation metrics are adopted in the multi-label

classification experiment include Hamming Loss (HL), One-Error (OE), Coverage (Co), Ranking Loss (RL), and Average Precision (AP) [21]. Hamming loss is a label-based metric, and the rest can be divided into ranking-based metrics.

We utilize word2vec[1] to train the word vectors on the 1.1 million Chinese Weibo corpora provided by NLPCC2017[2]. The dimension of word embedding vectors is set as 200 and the size of hidden layer is set as 100. All out-of-vocabulary words are initialized to zero. The maximum sentence length is 90. All weight matrices and bias are randomly initialized by a uniform distribution $U(-0.01, 0.01)$. TensorFlow is used to implement our neural network model. In model training, learning rate is set as 0.005, L_2-norm regularization is set as 1e−4, the parameter λ_1 in the emotion constraint term is set as 1e−3. We use the stochastic gradient descent (SGD) algorithm and Adam update rule with shuffled mini-batch for parameter optimization.

3.2 Comparison with Traditional Multi-label Learning Models

In this section, we compare JBNN with six strong multi-label learning models for multi-label emotion classification, namely EDL [22], ML-KNN [21], Rank-SVM [21], MLLOC [3], ECC [11], LIFT [19]. For each algorithm, ten-fold cross validation is conducted.

Table 1 shows the experimental results of the proposed method in comparison with the six strong multi-label learning methods. The two-tailed t-tests with 5% significance level are performed to see whether the differences between JBNN and the compared models are statistically significant. We can find that the MLLOC method is the worst, and the ECC method performs better than MLLOC. The experimental performance of MLKNN and LIFT is similar, while the performance of RankSVM is slightly worse than them. Among these traditional multi-label learning models, EDL performs the best. However, our model improves the EDL method with an impressive improvement in all kinds of evaluation metrics, *i.e.*, 10.02% reduction in RL, 4.60% reduction in HL, 12.04%

Table 1. Experimental results in comparison with traditional multi-label learning methods (mean ± std). '↓' means 'the smaller the better'. '↑' means 'the larger the better'. Boldface highlights the best performance. '•' indicates significance difference.

Algorithm	RL(↓)	HL(↓)	OE(↓)	Co(↓)	AP(↑)
ECC [11]	0.3281 ± 0.0659•	0.1812 ± 0.0940•	0.6969 ± 0.0598•	2.7767 ± 0.0876•	0.5121 ± 0.0892•
MLLOC [3]	0.4742 ± 0.0734•	0.1850 ± 0.0659•	0.6971 ± 0.0924•	3.6994 ± 0.0764•	0.4135 ± 0.0568•
ML-KNN [21]	0.2908 ± 0.0431•	0.2459 ± 0.0781•	0.5339 ± 0.0954•	2.4480 ± 0.0981•	0.5917 ± 0.0742•
Rank-SVM [21]	0.3055 ± 0.0579•	0.2485 ± 0.0458•	0.5603 ± 0.0921•	2.5861 ± 0.0777•	0.5738 ± 0.0892•
LIFT [19]	0.2854 ± 0.0427•	0.1779 ± 0.0597•	0.5131 ± 0.0666•	2.4267 ± 0.0492•	0.5979 ± 0.0891•
EDL [22]	0.2513 ± 0.0560•	0.1772 ± 0.0568•	0.5239 ± 0.0945•	2.1412 ± 0.0235•	0.6419 ± 0.0235•
JBNN (Our approach)	**0.1511 ± 0.0030**	**0.1312 ± 0.0009**	**0.4035 ± 0.0073**	**1.7864 ± 0.0193**	**0.7171 ± 0.0041**

[1] https://code.google.com/archive/p/word2vec/.
[2] http://www.aihuang.org/p/challenge.html.

reduction in OE, 35.48% reduction in Co and 7.52% increase in AP. In short, it can be observed that our JBNN approach performs consistently the best on all evaluation measures. The improvements are all significant in all situations.

3.3 Comparison with Two Types of Neural Networks (BRNN and TDNN)

These models usually utilize neural networks to automatically extract features of sentence and obtain final results. In this section, we compare our proposed JBNN with two major neural networks for multi-label classification, namely BRNN and TDNN, with multi-label classification performance and computational efficiency. We implement all these approaches based on the same neural network infrastructure, use the same 200-dimensional word embeddings, and run them on the same machine. The details of implement are as follows:

- **BRNN** is implemented by constructing multiple binary neural networks, as shown in Fig. 1(a), based on Bi-LSTM and attention mechanism.
- **TDNN** is implemented using the method in [13], which used a neural network based method to train one multi-class classifier and c binary classifiers to get the probability values of the c emotion labels, and then leveraged Calibrated Label Ranking (CLR) method to obtain the final emotion labels.

Classification Performance. In Table 2, we report the performance of JBNN, BRNN and TDNN models. From this table, we can see that our JBNN model performs significantly better than BRNN among all five kinds of evaluation metrics. Compared with the TDNN, our JBNN model is much better in Ranking Loss, Hamming Loss, One-Error, Average Precision. In general, our JBNN model performs better than both BRNN and TDNN models. The improvements according to two-tailed t-test are significant.

Computational Efficiency. We also report the size of parameters and runtime cost of BRNN, TDNN and JBNN in Table 3. From Table 3, we can find that our JBNN model is much simpler than BRNN and TDNN. For example, our JBNN model only has 0.28 M parameters, while BRNN has 2.53M parameters and TDNN has 2.81M parameters. As for runtime cost, we can see that BRNN and TDNN are indeed computationally expensive. Our JBNN model is almost 8 times faster than BRNN and 9 times faster than TDNN in model training. In summary, our JBNN model has significantly priority against BRNN and TDNN in computation efficiency.

Table 2. Experimental results in comparison with two types of neural networks methods (mean ± std). '↓' means 'the smaller the better'. '↑' means 'the larger the better'. Boldface highlights the best performance. '•' indicates significance difference.

Algorithm	RL(↓)	HL(↓)	OE(↓)	Co(↓)	AP(↑)
BRNN	0.1612 ± 0.0051•	0.1346 ± 0.0015•	0.4243 ± 0.0073•	1.8779 ± 0.0371•	0.7017 ± 0.0054•
TDNN	0.1532 ± 0.0040•	0.1334 ± 0.0013•	0.4148 ± 0.0098•	1.7922 ± 0.0299	0.7115 ± 0.0060•
JBNN	**0.1511 ± 0.0030**	**0.1312 ± 0.0009**	**0.4035 ± 0.0073**	**1.7864 ± 0.0193**	**0.7171 ± 0.0041**

Table 3. Computational Efficiency of different neural networks. Params means the number of parameters, while Time cost means runtime (seconds) of each training epoch.

Algorithm	Params(↓)	Time cost(↓)
BRNN	2.53M	265 s
TDNN	2.81M	305 s
JBNN	**0.28M**	**35 s**

4 Conclusion

In this paper, we have proposed a joint binary neural network (JBNN) model for multi-label emotion classification. Unlike existing multi-label learning neural networks, which either needs to train a set of binary networks separately (BRNN), or although model the problem within a multi-class network, an extra threshold function is needed to transform the multi-class probabilities to multi-label outputs (JDNN), our model is an end-to-end learning framework that integrates representation learning and multi-label classification into one neural network. Our JBNN model is trained on a joint binary cross entropy (JBCE) loss. Furthermore, the label relation prior is also incorporated to capture the correlation between emotions. The experimental results show that our model is much better than both traditional multi-label emotion classification methods and the representative neural network systems (BRNN and TDNN), in both multi-class classification performance and computational efficiency.

Acknowledgments. The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
3. Huang, S.J., Zhou, Z.H., Zhou, Z.: Multi-label learning by exploiting label correlations locally. In: AAAI, pp. 949–955 (2012)
4. Lenc, L., Král, P.: Deep neural networks for Czech multi-label document classification. arXiv preprint arXiv:1701.03849 (2017)
5. Li, S., Huang, L., Wang, R., Zhou, G.: Sentence-level emotion classification with label and context dependence. In: ACL, pp. 1045–1053 (2015)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)

7. Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., Fürnkranz, J.: Large-scale multi-label text classification — revisiting neural networks. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8725, pp. 437–452. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44851-9_28
8. Plutchik, R.: Chapter 1 - a general psychoevolutionary theory of emotion. Elsevier Inc. (1980)
9. Quan, C., Ren, F.: Sentence emotion analysis and recognition based on emotion words using ren-cecps. Int. J. Adv. Intell. **2**(1), 105–117 (2010)
10. Read, J., Perez-Cruz, F.: Deep learning for multi-label classification. arXiv preprint arXiv:1502.05988 (2014)
11. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5782, pp. 254–269. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04174-7_17
12. Spyromitros, E., Tsoumakas, G., Vlahavas, I.: An empirical study of lazy multilabel classification algorithms. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 401–406. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87881-0_40
13. Wang, Y., Feng, S., Wang, D., Yu, G., Zhang, Y.: Multi-label Chinese microblog emotion classification via convolutional neural network. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds.) APWeb 2016. LNCS, vol. 9931, pp. 567–580. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45814-4_46
14. Wang, Y., Pal, A.: Detecting emotions in social media: a constrained optimization approach. In: IJCAI, pp. 996–1002 (2015)
15. Xu, G., Lee, H., Koo, M.W., Seo, J.: Convolutional neural network using a threshold predictor for multi-label speech act classification. In: BigComp, pp. 126–130 (2017)
16. Xu, J., Xu, R., Lu, Q., Wang, X.: Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context. In: CIKM, pp. 2455–2458 (2012)
17. Yan, J.L.S., Turtle, H.R.: Exposing a set of fine-grained emotion categories from tweets. In: IJCAI, p. 8 (2016)
18. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: HLT-NAACL, pp. 1480–1489 (2016)
19. Zhang, M.L., Wu, L.: Lift: multi-label learning with label-specific features. IEEE Trans. Pattern Anal. Mach. Intell. **37**(1), 107–120 (2015)
20. Zhang, M.L., Zhou, Z.H.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. **18**(10), 1338–1351 (2006)
21. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014)
22. Zhou, D., Zhang, X., Zhou, Y., Zhao, Q., Geng, X.: Emotion distribution learning from texts. In: EMNLP, pp. 638–647 (2016)