Deep MIML Network*

Ji Feng and **Zhi-Hua Zhou**
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China
{fengj, zhouzh}@lamda.nju.edu.cn

Abstract

In many real world applications, the concerned objects are with multiple labels, and can be represented as a bag of instances. Multi-instance Multi-label (MIML) learning provides a framework for handling such task and has exhibited excellent performance in various domains. In a MIML setting, the feature representation of instances usually has big impact on the final performance; inspired by the recent deep learning studies, in this paper, we propose the DeepMIML network which exploits deep neural network formation to generate instance representation for MIML. The sub-concept learning component of the DeepMIML structure reserves the instance-label relation discovery ability of MIML algorithms; that is, it can automatically locating the key input patterns that trigger the labels. The effectiveness of DeepMIML network is validated by experiments on various domains of data.

Introduction

In many real-world applications, an object of interest has its inherent structure and it can be represented as a bag of instances, and multiple labels are associated on the bag level. For example, in text categorization, each document may have sentences as instances and multiple labels are assigned on document level. Multi-Instance Multi-Label (MIML) (Zhou and Zhang 2006; Zhou et al. 2012) provides a framework for handling these kinds of tasks.

Concretely, in a MIML perspective, the training data $\{(X_1, Y_1), ..., (X_m, Y_m)\}$ consists of m bags of instances, where each bag X_i can be represented as z_i instances such as $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}...\boldsymbol{x}_{i,z_i}\}$. The output Y_i is a subset of all possible labels $\{y_1, y_2, ...y_L\}$ where L is the number of possible single labels. Many MIML algorithms such as (Zhou et al. 2012; Briggs, Fern, and Raich 2012; Yang, Jiang, and Zhou 2013; Nguyen et al. 2014; Huang, Gao, and Zhou 2014) have been proposed during the past years and have been successfully applied in different domain of tasks such as image classification, text categorization, video annotation, gene function prediction, ecosystem protection, etc. (Xu, Xue, and Zhou 2011; Zhou et al. 2012; Surdeanu et al. 2012; Briggs, Fern, and Raich 2013; Wu, Huang, and Zhou 2014)

*Supported by the 973 Program (2014CB340501) and NSFC (61333014).

Most previous MIML studies assumed that the instances were given in advance, or generated by some manually designed *instance generators* which extract instances from the raw data. Recently, an empirical study (Wei and Zhou 2016) on image tasks exhibit that there is no manually designed instance generator dominating others. Considering that representation learning techniques have beat manually feature engineering in various domains, it naturally motivates us to try to facilitate MIML with automatic representation learning, though larger amount of data are required for this purpose.

In this paper, we propose the DeepMIML network. As its name suggested, this is a deep neural network model for MIML. Born with the representation learning ability of deep models, in DeepMIML we do not need to use another instance generator to generate instance descriptions. Instead, the model itself will accomplish the instance representation generation and the successive learning process. Moreover, with a carefully designed sub-concepts layer, an apparent advantage of MIML, i.e., discovering the latent relation between input patterns and output semantic labels (Zhou et al. 2012; Li et al. 2012) has been reserved. Such a layer can also be plugged into other types of network structures, such as CNNs, to endow them with the ability of pattern-label relation discovery ability. The effectiveness of DeepMIML are validated in experiments.

The rest of the paper starts from a brief review of related work. Then we present the DeepMIML network, followed by experiments and conclusion.

Related Works

During the past few years, many successful algorithms for MIML have been proposed, to name a few, Li *et al.* tried (2012) to model what instances trigger what labels by considering the shared patterns across relevant labels; Briggs *et al.* (2012) proposed rank-loss support instance machines for MIML instance anotation; Briggs *et al.* (2013) considered the problem of predicting instance labels while learning from data labeled only at the bag level by using a new regularized rank-loss objective; Huang *et al.* (2014) proposed a fast MIML algorithm by exploiting label relations with shared space and discovering sub-concepts for complicated labels; Pham *et al.*(2015) used a discriminative probabilistic model to discover novel class instances in a MIML setting.

Most previous MIML approaches treated the representations for instances as given, and the scalability to large training data requires to be enhanced.

In recent years, deep learning has demonstrated its ability in learning representations from raw data. In particular, deep covolutional neural networks such as (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014) has showed its superior performances on image classification tasks. LSTM (Hochreiter and Schmidhuber 1997) based models, has been successfully applied to text data; for instance, a pre-trained skip-thought model (Kiros et al. 2015; Zhu et al. 2015) can be used as an off-the-shelf encoder to produce high quality sentence representations. Note that most of the focus in the deep learning community lies in representation learning, it would be desirable if advantages of some other machine learning paradigms, such as the pattern-label relation discovery of MIML, can be incorporated.

There are studies using deep learning techniques on multi-label problems such as (Wei et al. 2014; Zeng et al. 2015; Wu et al. 2015; Lin et al. 2016; Wang et al. 2016). Most were domain specific solutions, and did not touch instance generators and pattern-label relation discovery.

The Proposed Approach

In this section, we will first introduce the 2D sub-concept layer which models the relationships between a single instance and output labels. Secondly, we will extend this idea into a 3D sub-concept layer which can be used in a MIML perspective. Finally, we introduce the Deep MIML network and discuss its ability of discovering instance label relationships.

2D Sub-Concept Layer for Single Instance

In semantic-rich tasks, labels may deliver complicated information, and thus, a direct modeling might be difficult. Instead, we propose a new 2-Dimensional neural network layer, the *sub-concept layer* as we call it, which can be trained to model *the matching scores between an instance and sub-concepts* for each label. Concretely, once the representation for an instance $\boldsymbol{x}$ is obtained, we propose a fully connected 2D layer (sub-concept layer) of size $K * L$. Formally, for a given instance vector $\boldsymbol{x}$, the (i, j)-th node in the 2D sub-concept layer represents the matching score between this instance $\boldsymbol{x}_{p,q}$ and the i-th sub-concept for the j-th label. That is, the (i, j)-th node has the following form of activation:

$$c_{i,j} = f(\boldsymbol{w}_{i,j}\boldsymbol{x} + b_{i,j}) \tag{1}$$

Here, $f(.)$ is the activation function and the weight vector $\boldsymbol{w_{i,j}}$ can be interpreted as the matching template for the i-th sub-concept of the j-th label. The activation function we chose here is the Rectified Linear Unit (ReLU) of the form:

$$f(z) = max(0, z) \tag{2}$$

This 2D sub-concept layer, as the name suggests, tries to model the matching scores between a single instance and all the sub-concepts for all the labels. To make a label level prediction, an immediate column-wise pooling operation on this 2D layer will produce a $K * 1$ scoring layer, each entry is thus the matching score between the instance $\boldsymbol{x}$ and the output label accordingly.

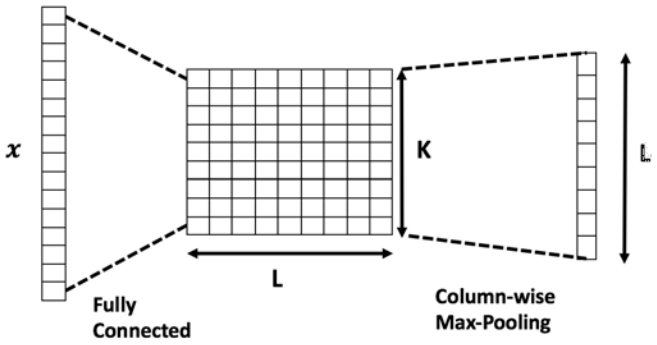

Figure 1: Illustration of a 2D Sub-concept layer

The pooling operation not only extract label predictions but also can have some robustness on the number of sub-concepts. That is, when a label has fewer than K sub-concepts, a max-pooling operation will eliminate this over-assignment on sub-concepts.

Compared with other commonly seen network structures, there are some points worth highlighting:

- Unlike the usual 2D convolutional feature maps (with local connective filters) , the 2D sub-concept layer we proposed here is fully connected with the input instance vector and the activations can be represented as the matching scores between a sub-concept for each label and the instance. Also notice that the weights are different for each node, whereas a shared weight is used in conv-layers.
- Unlike the usual 1D fully connected layer, the 2D sub-concept layer is arranged in an interpretable way. In other words, the layer is a direct consequence when modeling such relationships such that each column is a score vector for each label, and each element in the column vector is the matching score for the sub-concepts for that particular label. This makes the layer much more intuitively easy to explain, and, more importantly, it can be used to discover the instance-label relationships (will be explained in later sections).
- The max-pooling operation was usually used for down-sampling to reduce the total number of parameters. However, the max operation we are using here is to locate the maximum matching score which happens can be expressed in a max-pooling layer.

Briefly speaking, the 2D sub-concept layer followed by a pooling layer captures the relationship between an input pattern and the final matching score for each label. In addition, the matching score for each sub-concept can be further used in instance-label relationship discovery, which will be discussed in later sections.

3D Sub-Concept Layer for Multiple Instances

When the input is presented in one bag of instances (here we assume each bag has equal number of instances, for bags

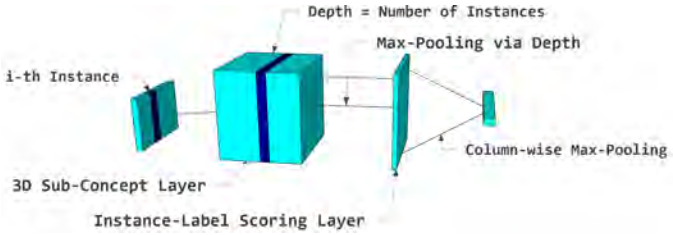

Figure 2: 3D Sub-concept layer and Instance-Label scoring layer. Each instance is connected with its corresponding sub-concept layer only. The resulting three-dimensional tensor has depth equals the number of input instances.

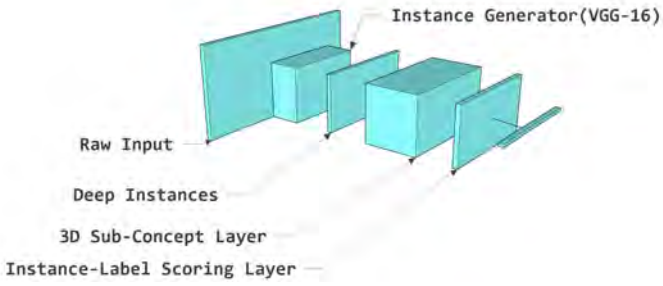

Figure 3: DeepMIML Network

with different number of instances, zero paddings may be applied), we can generalize the idea of 2D sub-concept layer into a MIML perspective. The basic idea is to extend the 2D sub-concept layer into a 3D tensor layer by *stacking* multiple 2D layers , each "slice" of the tensor is the 2D sub-concept layer for each instance, as described in the previous section.

In other words, given a bag of instances X_i, we construct the 2D sub-concept layer for each instance x_{ki}, and *stack* these 2D layers into a 3D tensor. The depth of the tensor equals to the number of instances in the input bag. That is, the activation of the (i, j, k)-th node represents the matching score of the i-th sub-concept of the j-th label for the k-th instance $\boldsymbol{x}_{k,i}$ in bag X_i. Note that the matching weights for each instance on the same sub-concepts is different.

To explore instance label relationships, we conduct the pooling operation twice. Concretely, we first conduct a max-pooling vertically on the 3D tensor, and the resulting layer is a 2-Dimensional layer of size $L * M$. Note that there is a clear interpretation for this 2D pooling layer: each node at position (i, j) models the matching score for instance i on label j. We thus refer to this first pooling layer as *Instance-Label Scoring Layer*.

Secondly, another pooling operation on the 2D Instance-Label Scoring Layer will produce a 1D layer of size $L * 1$. This can be interpreted as the matching scores for labels on *bag* level: each entry j models the matching score for the j-th label on the whole input bag.

In short, to get the $L * 1$ layer from the 3D tensor layer, we conducted the pooling operation twice: a vertical pooling followed by a horizontal pooling.

The reason we conduct the pooling operation twice (instead of a single 2D pooling) is because the intermediate instance-label scoring layer has an unique interpretations on its own, and by examine the values on this layer can help to understand the instance label relationships. A detailed discussion will be presented in later sections.

The DeepMIML Network

Previous sections has showed how to model the MIML framework by exploring sub-concepts for labels using the language of neural network, under the assumption of having input representations as given. In this section, we will first introduce how to learn a bag of instances from raw data, and finally introduce an end-to-end Deep MIML network which connect everything we've discussed.

Recent advances in deep learning has made deep neural networks particularly good at learning feature representations. For example, it is a well known technique for encoding an input image as a dense 1D vector by extracting the activations from the last fully connected layer (FC7) from a trained CNN.

However, such method treat the input as a whole, and thus the distributed dense 1D representation for the input image cannot reveal local information (instances) very well. Here we do things a bit different as follows: In a deep convolutional neural network structure, the layer before the last fully connected 1D layers is a 3D convolutional tensor layer, usually of shape $14 * 14 * 512$ (e.g. VGG-16 Net). Inside this conv-layer, each of of the $14 * 14$ vector with dimension $512 * 1$ can be treated as one instance representation of the input image. Thus, in a MIML perspective, we use the representations in the conv-layer rather than the FC layer in order to obtain a bag of instance representations.

Now, we are ready to introduce the Deep MIML network. It is a novel network structure that automatically generates bag of instances from raw input, learns a scoring function for each sub-concept of each label on instance level and makes final predictions on the bag level.

Concretely, the raw input is first fed into an *instance generator* device, of which is dependent on the domain of interest. For image tasks, a deep convolutional neural network structure just before the FC layer will serve the job well. Then, a 3D sub-concept layer followed by two pooling layers are directly applied to the instance generator, as described in the previous section. Finally, a fully connected layer with size equal to the number of labels is appended at last. The loss function we choose here is the mean binary cross-entropy. During training time, we use stochastic gradient descent with dropout. See Figure 3 as an illustration.

There are three points worth highlighting:

- Firstly, a more sophisticated instance generator may be used for specific tasks. For example, for image tasks, some proposal based method (creating some small bounding boxes before fed into conv-nets) can be applied first to get an even better representation for instance. An pre-trained LSTM based encoder, on the other hand, can be used in NLP tasks.
- Secondly, there are situations when we do not have access

to the raw data, and the input has already been encoded into bags of instances. In this case, we can still use the DeepMIML framework by directly project them onto the 3D sub-concept layer.

- Finally, the network can be easily degenerated into a single-instance multi-label or multi-instance single-label case, by changing the dimension of the sub-concept layer accordingly.

Instance-Label Relation Discovery

Instance-Label Relation Discovery is the discovery process of locating the key instance pattern that triggers the output labels. It is different from *instance annotation*, since annotation cares about assigning correct labels to instances no matter if the instance is the truly trigger for the output label. Therefore, a good performance of instance annotation does not necessarily lead to a good performance in the instance-label relation discovery.

The instance label relation discovery is a built-in functionality of DeepMIML networks. The first pooling layer (namely the *instance-label scoring layer*) after the 3D sub-concept layer will produce a matching score across all instances for all labels. By examine these activation scores in this instance-label scoring layer, it is a straightforward routine to conduct instance-label relation discovery.

Concretely, on MS-COCO data-sets, each input image is transformed into a bag of 196 instances(via a VGG-16 net) and there are 80 candidate labels can be tagged with. The corresponding instance-label scoring layer of size $196 * 80$ give us all the matching scores between each instance and each label. By examine the activations, we can easily discover the relation between instances and labels.

In addition, we can also detect which key instance triggers one particular label by back-track the location of the instance with the highest matching score in the 2D pooling layer accordingly. For a VGG-16 architecture, the formula of locating the center pixel for the key instance can be easily derived:

$$(x, y) = (conv_x * 16 + 8, conv_y * 16 + 8) \qquad (3)$$

The $conv_x$, $conv_y$ is the 2D indexing in the 14*14 instance bag. By doing so, we can achieve similar effect with the attention mechanism. See Figure 4 for such an result.

Experiments

To validate the effectiveness of the proposed network structure, we performed experiments on both text data and image data with considerable scale in terms of training sizes. The goal here is to show that Deep MIML network is a general network structure suitable for many MIML problems across domains and can be easily deployed in various tasks with slight modifications.

In addition, we showed that by simply plug the 3D sub-concept layer (and the pooling layers followed by it) into the VGG-net structure, we can have a performance gain in both accuracy and extra benefits such as label-instance relationship discovery.

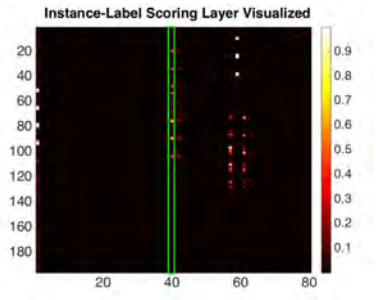

Figure 4: Instance-Label Matching Plot. For a given input image to be predicted, 196 (14*14) instances is extracted from VGG-16 net, and there are 80 candidate labels in the MS-COCO dataset. Activations for the instance-label scoring layer can be visualized and interpreted as follows: Each column is the matching scores between all the instances and one particular label. The highlighted green column, for example, is the matching scores for the label "Bottle" between all the instances. Likewise, each row can be interpreted as the matching scores between one particular instance and all the possible labels.

To make a fair comparison with the traditional state-of-art MIML algorithms, and to validate the effectiveness of the sub-concept layer, we conducted experiments on data-sets without deep features. Specifically, we used two well-known MIML benchmark data-sets, namely MIML News and MIML Scene (Zhou et al. 2012) , which the raw data has already been preprocessed into bags of instance format, and we compared the experimental results with state-of-art MIML algorithms.

Concretely, we report the experimental results as follows:

- DeepMIML for Text Tasks: We performed experiments on 2016 Yelp dataset challenge [1]. Each review belongs to one or more categories (such as "restaurant","Thai Food") and we extract 100 categories with the reviews been tagged with.
- DeepMIML for Image Tasks: We use MS-COCO dataset consists of 82,783 images for training with total 80 labels. To get one bag of instance per image, we used a pre-trained VGG-16 net up-to the last convolutional layer.
- DeepMIML for non-deep features: We compared our results with the state-of-art MIML algorithms on some traditional MIML data-sets.[2] [3] Since each instance has already been hand coded into feature vectors, thus we also used the same input data as representations, for a fair comparison.

We implemented the model via Keras ($keras.io$), an open-source python library for deep learning which allow users to customize their own network structure and layers. We also used 2 Nvidia Titan-X GPU to speed up training time.

[1] yelp.com/dataset_challenge

[2] http://lamda.nju.edu.cn/files/miml-text-data.rar

[3] http://lamda.nju.edu.cn/files/miml-image-data.rar

Results on Text Data

For MIML tasks for text, we conducted the experiment on the 2016 Yelp dataset challenge. Specifically, We extracted 19934 reviews, each review belongs to one or more categories as labels and there are 100 categories in total. We split the dataset into training and testing set, with the split ratio of 0.7.

To encode each input review into a multi-instance representation, we used a pre-trained skip-thought model (Kiros et al. 2015) as an out-of-shelf encoder for each sentence (as instances). The input can be a sequence of words of any length, and the output is a 4800 dimensional vector. Table 1 shows a given query (not seen in the dataset) and its nearest / farthest neighbors in the yelp dataset.

Table 1: Sample Query sentence

Query Sentence	The beef is good.
Nearest Sentences	The curries are nice too. The calamari is good. The BBQ is great. The food is great the set up is nice.
Farthest Sentences	Nope nope nope. Disappointed. Not coming back. Dislike.

We then divide each review into 10 sequences of words, each sequence contains one sentence. If one review has more than 10 sentences, then the last part contains all the remaining words. With the bag of (deep) instance representations in hand, we directly applied the 3D sub-concept layer (and with the following pooling layers), as described in the previous section. We used mean binary cross-entropy as loss function and used SGD with dropout rate of 0.5. The only hyper-parameters here is K, the number of sub-concepts. During validating process, we found varying K does not affect the performance a lot and we report the result with K equal to 4.

For a comparison, we still use skip-thought to encode the whole review into one dense representation with dimension 4800, then we perform a soft-max and a MLP (two hidden layers of size 1024 followed by 512 with ReLU activation) as benchmark comparisons. The evaluation metrics we used here are commonly used criteria for multi-label tasks, as studied in (Zhou et al. 2012). The experimental results are summarized in Table 2.

Table 2: Experimental Results on Yelp dataset. mAP denotes *mean Average Precision*. ↑ / ↓ indicates the larger/smaller, the better. The • symbol means the DeepMIML Network performs better than the corresponding method.

	$mAP \uparrow$	$ranking\ loss \downarrow$
Softmax	•0.313	•0.083
MLP	•0.325	•0.080
Our Method	0.330	0.078

Figure 5: Sample test image predictions and the attention mechanism achieved by sub-concept layer.

Results on Image Data

For image task, we conducted our experiments on Microsoft COCO dataset (Lin et al. 2014). The MS-COCO dataset contains 82,783 images for training and 40,504 images for testing. Each image provides annotation tags of 80 labels. Here we only used the annotated text on image level as labels and used a pre-trained VGG-16 up to the last convolutional layer as the instance generator. The training time is less than 30 minus if we take the pre-trained VGG-16 fixed.

Figure 5 is an illustration of the predictions obtained by discovering the instance-label relation on the test set. An *attention mechanism* can be easily achieved via the sub-concept layer, as discussed in the previous section.

Table 3 showed the comparison results. Compared with using a vanilla VGG network, we indeed have a performance gain in terms of accuracy by simply adding one extra sub-concept layer.

Table 3: Experimental Results on MS-COCO dataset

	mAP	$HammingLoss$	$F1$
VGG-16	57 %	0.025	0.650
CNN-RNN	61.2 %	–	0.678
Our Method	60.5%	0.021	0.637

Compared to the more sophisticated CNN-RNN model, our method showed sub-optimal performance. This is because the instance generator we use here is very simple and straight forward and tiny object such as parking meter cannot be efficiently encoded into instances. On the other hand, CNN-RNN is a good state-of-art model for image tasks only and cannot be easily applied in other non-CV tasks easily. Again, the goal here is to show the DeepMIML network is general enough for tasks across domain with minimal mod-

ifications. In addition, our model can be easily degenerated into multi-instance single-label framework (by replacing the 3D sub-concept layer into 2D version), whereas CNN-RNN method cannot be effectively deployed.

DeepMIML Applied to Common Instances

In this section, we compare our method with some state-of-art MIML algorithms, namely, KISAR (Li et al. 2012), MIML SVM, MIML KNN , MIML RBF and MIML Boost (Zhou et al. 2012).

For a fair comparison, we used the benchmark dataset reported in these prior works by Zhou *et. al* (2012) among others, which has already been preprocessed using tf-idf (MIML News data) and SBN features (Wei and Zhou 2016) on patches (MIML Scene data). Therefore, we directly projected these instances onto the 3D sub-concept layer, and the remaining network layers are the same as previous ones. Note that these dataset has relatively small in both the number of instances and the number of possible labels.

The evaluation metrics we used here are the same as those algorithms reported in their corresponding papers. Details of the evaluation measures can be found in (Zhang and Zhou 2014).

Table 4: Experimental Results on MIML datasets. h.l, c.o, r.l denote *hamming loss*, *coverage* and *ranking loss*, respectively. ↑ / ↓ indicates the larger/smaller, the better.

MIML News Data			
	h.l.↓	c.o ↓	r.l ↓
Our Method	**0.160**	**0.890**	**0.157**
KISAR	•0.167	•0.928	•0.162
MIML SVM	•0.184	•1.039	•0.190
MIML KNN	•0.172	•0.944	•0.169
MIML RBF	•0.169	•0.950	•0.169
MIML Boost	•0.189	•0.947	•0.172
MIML Scene Data			
	h.l. ↓	c.o ↓	r.l ↓
Our Method	**0.026**	**0.261**	**0.016**
KISAR	•0.032	•0.278	•0.019
MIML SVM	•0.044	•0.373	•0.034
MIML KNN	•0.063	•0.489	•0.051
MIML RBF	•0.061	•0.481	•0.052
MIML Boost	•0.053	•0.417	•0.039

Conclusion

In this paper, we propose a general deep model capable of solving MIML problems across various domains. In contrast to previous MIML studies that rely on manually designed instance generators, our proposed DeepMIML, born with the representation learning ability of deep models, is able to learn the instance description automatically. The sub-concept layer, which is easy to be incorporated into other deep models, enables DeepMIML to inherit the ability of MIML for discovering the relation between input patterns and output semantic labels. Experiments on various datasets demonstrated the superior performance of DeepMIML.

References

Briggs, F.; Fern, X. Z.; Raich, R.; and Lou, Q. 2013. Instance annotation for multi-instance multi-label learning. *ACM Transactions on Knowledge Discovery from Data* 7(3):14.

Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for MIML instance annotation. In *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 534–542.

Briggs, F.; Fern, X. Z.; and Raich, R. 2013. Context-aware MIML instance annotation. In *13th IEEE International Conference on Data Mining*, 41–50.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Huang, S.-J.; Gao, W.; and Zhou, Z.-H. 2014. Fast multi-instance multi-label learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1868–1874.

Kiros, R.; Zhu, Y.-K.; Salakhutdinov, R.; Zemel, R. S.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28*, 3294–3302.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 1106–1114.

Li, Y.-F.; Hu, J.-H.; Jiang, Y.; and Zhou, Z.-H. 2012. Towards discovering what patterns trigger what labels. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 1012–1018.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision*, 740–755.

Lin, Y.-K.; Shen, S.-Q.; Liu, Z.-Y.; Luan, H.-B.; and Sun, M.-S. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2124–2133.

Nguyen, C.-T.; Wang, X.-L.; Liu, J.; and Zhou, Z.-H. 2014. Labeling complicated objects: Multi-view multi-instance multi-label learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2013–2019.

Pham, A. T.; Raich, R.; Fern, X. Z.; and Arriaga, J. P. 2015. Multi-instance multi-label learning in the presence of novel class instances. In *Proceedings of the 32nd International Conference on Machine Learning*, 2427–2435.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 455–465.

Wang, J.; Yang, Y.; Mao, J.-H.; Huang, Z.-H.; Huang, C.; and Xu, W. 2016. CNN-RNN: A unified framework for multi-label image classification. *CoRR* abs/1604.04573.

Wei, X.-S., and Zhou, Z.-H. 2016. An empirical study on image bag generators for multi-instance learning. *Machine Learning* 105(2):155–198.

Wei, Y.-C.; Xia, W.; Huang, J.-S.; Ni, B.-B.; Dong, J.; Zhao, Y.; and Yan, S.-C. 2014. CNN: single-label to multi-label. *CoRR* abs/1406.5726.

Wu, J.-J.; Yu, Y.-N.; Chang, H.; and Yu, K. 2015. Deep multiple instance learning for image classification and auto-annotation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3460–3469.

Wu, J.-S.; Huang, S.-J.; and Zhou, Z.-H. 2014. Genome-wide protein function prediction through multi-instance multi-label learning. *IEEE/ACM Trans. Comput. Biology Bioinform.* 11(5):891–902.

Xu, X.-S.; Xue, X.-Y.; and Zhou, Z.-H. 2011. Ensemble multi-instance multi-label learning approach for video annotation task. In *Proceedings of the 19th ACM International Conference on Multimedia*, 1153–1156.

Yang, S.-J.; Jiang, Y.; and Zhou, Z.-H. 2013. Multi-instance multi-label learning with weak label. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 1862–1868.

Zeng, D.-J.; Liu, K.; Chen, Y.-B.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 17–21.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.* 26(8):1819–1837.

Zhou, Z.-H., and Zhang, M.-L. 2006. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19*, 1609–1616.

Zhou, Z.-H.; Zhang, M.-L.; Huang, S.-J.; and Li, Y.-F. 2012. Multi-instance multi-label learning. *Artificial Intelligence* 176(1):2291–2320.

Zhu, Y.-K.; Kiros, R.; Zemel, R. S.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 15th IEEE International Conference on Computer Vision*, 19–27.