Multimed Tools Appl (2018) 77:29799–29810
https://doi.org/10.1007/s11042-018-5772-4

Few-shot learning for short text classification

Leiming Yan[1] · Yuhui Zheng[2] · Jie Cao[3]

Received: 26 September 2017 / Revised: 31 January 2018 / Accepted: 9 February 2018 /
Published online: 21 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Due to the limited length and freely constructed sentence structures, it is a difficult classification task for short text classification. In this paper, a short text classification framework based on Siamese CNNs and few-shot learning is proposed. The Siamese CNNs will learn the discriminative text encoding so as to help classifiers distinguish those obscure or informal sentence. The different sentence structures and different descriptions of a topic are viewed as 'prototypes', which will be learned by few-shot learning strategy to improve the classifier's generalization. Our experimental results show that the proposed framework leads to better results in accuracies on twitter classifications and outperforms some popular traditional text classification methods and a few deep network approaches.

Keywords Convolutional neural networks · Deep learning · Few-shot learning · Text classification

1 Introduction

Few-shot learning refers to a new kind of learning techniques which utilize only a few labeled samples for model training. Recently, some few-shot learning methods, including one-shot

✉ Leiming Yan
lmyan@nuist.edu.cn

Yuhui Zheng
zhengyh@vip.126.com

Jie Cao
cj@nuist.edu.cn

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, China

[2] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, China

[3] School of Mathematical &Statistics, Nanjing University of Information Science & Technology, Nanjing, China

learning and zero-shot learning have been proposed to reduce the number of necessary labeled samples and succeeded in visual object classification [3]. In contrast, recent great success achieved by deep learning approaches relies on large amounts of labeled samples in image classification, machine translation, and speech modeling [25]. However, in reality, it is difficult to obtain sufficient labeled samples for machine learning approaches because the labeling cost is prohibitively expensive or they do not occur frequently enough to be collected. Few-shot learning therefore has shown great potential in saving the labeling efforts.

Some few-shot learning approaches achieved great performance in vision domain, where a classifier could quickly generalize after trained by very few samples from each class. However, it seems more complicated in natural language processing and classification, especially for short text classification. Due to the length limit, abbreviations are commonly used in short text and message, such as twitter, and the sentence is normally informal and freely constructed with poor grammar and lots of misspellings. The semantic space of a language is generally believed as a high dimension space, and semantic features or word embedding are believed to be beneficial in many NLP tasks, such as named entity recognition, parsing and sentiment analysis [30, 35]. Most of achievements in short text classification show that more labelled samples can improve the performance of classifiers [24].

Unfortunately, it seems that thousands of labelled samples, even millions of labelled samples, could still not contain sufficient semantic information so as to make the classifiers obtain satisfying performance. We do not know how many labelled samples is enough for short text classifier training. Due to sparsity, sentences are perhaps too short to provide enough information for labelling or discriminating different sentences in the distribution feature space. Motivated by few-shot learning, we consider the short text classification as a few-shot learning task. Since we could not get enough or millions of samples to train classifiers, few-shot learning become our natural choice.

In this work, we draw inspiration from the metric learning based on deep neural features, such as Large margin nearest neighbor (LMNN) classification [39] and Deep convolutional neural networks (CNN) [17]. We propose a framework based on few-shot learning to improve performance of text classification.

In Section 2, we refer to some earlier work on training classifiers from only a few examples. In Section 3, we describe the Siamese CNN architecture which will be used to learn discriminative text encoding. In Section 4, we specify the 'episode' construction method and the few-shot learning procedure. In Section 5, we show and analyze the resulting classification performance of the framework we proposed and compare its performance with the related approaches.

2 Related work

There are many promising works that achieved state-of-the-art performance in image and vision classification domain, like semantic search or similarity search [8, 9], semantic annotation [41], Co-Saliency detection [12, 13, 42, 44–46], image copy-move forgery detection [21, 49, 50] and feature selection [47, 48]. Recently, some sparse coding and hash embedding technologies [6, 7, 10] have been proposed for data retrieval. Moreover, Wang et al. [38] and Chen et al. [4] respectively proposed novel image feature spaces based on quaternion to extract more helpful information from color images. Their results show a certain improvement in detection accuracy. Although quite a lot of research has been done, learning new concepts or classes with only a few samples is still one of the major challenges in deep learning tasks. Zero-shot learning and one-shot learning are the special examples of few-shot learning.

Zero-shot learning has the most restrictive premise because there are no training samples for model training. In the work of Lampert et al. [20], an attribute-based classification is introduced, and it performs object detection based on a human-specified high-level semantic attributed instead of training images. Zhang et al. [43] develop a max-margin framework to learn semantic similarity embedding, and use the seen class proportion as a similarity measurement between unseen classes. Some works make use of side information such as visual attributes or natural language semantics to define the relations between output visual classes. In the work of Jetley et al. [16], visual prototypical concepts are used as side information to draw inference on new unseen classes. Guo et al. [11] propose a novel one-step recognition zero-learning framework which can perform recognition by trained classifiers. The framework proposes to transfer samples from source classes with pseudo labels.

In one-shot learning, new concepts are learned from a single sample. Several one-shot learning works learn distance metrics from domains that are related to the target domain. Schroff et al. [28] propose a system FaceNet, which learns a mapping between image space and Euclidean space to directly measure the similarity. Some works take use of probabilistic model for analysis of handwritten digits. Lake et al. [19] present a Hierarchical Bayesian model based on compositionality and causality that can learn a wide range of natural visual concepts. Oriol et al. [36] propose a matching network framework based on metric learning technology, which uses an attention mechanism over a support set to predict the unlabeled query set. Rezende et al. [26] develop a new deep generative model, which combines the representational power of deep learning with the Bayesian reasoning. Koch et al. [18] propose a method for learning siamese neural networks for one-shot image recognition and achieve strong results.

Few-shot learning refers to learning new concepts from more than one but still a limited number of samples. Hariharan et al. [14] present a low-shot learning benchmark on complex images. They pre-learn a network for ImageNet dataset, then train new classifiers with a few samples. Eleni et al. [34] propose a few-shot learning method through an information retrieval lens, which extracts as much information as possible from each training batch by optimizing over all relative orderings. Some novel approaches based on ‘prototype’ are proposed. Jake Snell et al. [29] propose prototypical networks for the problem of few-shot classification. Prototypical networks are used to learn a metric space in which distances between prototype representations of each class. Hecht et al. [15] also present a deep network model, which use prototypes to produce local maps between two hidden layers. Blaes et al. [3] re-define a new ‘prototype’ concept, and propose a method to find prototypes in the global feature layers so as to obtain good classification results over image datasets. Sachin et al. [25] propose a LSTM based meta-learner model to learn the optimized parameters, which can be used to train another learner neural network classifier.

Some promising works based on deep learning in NLP tasks have been proposed. Mikolov et al. [23] introduced word vector training algorithm and a RNN based language model with applications to speech recognition. After that, some frameworks based on LSTM are proposed [5, 32, 37], which improve the performance of RNN obviously. Cooolll [33] is a deep learning system for twitter sentiment analysis. This system is ranked 2nd on the evaluation of SemEval2104 Task 9. However, in top 20 systems of SemEval2104 evaluation, there are only two systems which applied word embedding and deep learning approaches [24]. Yan et al. [40] propose a twitter sentiment analysis method based on autoencoder which performs better than traditional bag-of-words approaches.

However, methods based on word embedding do not outweigh those traditional methods in short message and twitter classification. This paper aims to propose an efficient frame for short text classification.

3 Deep network architecture

Here, we choose the Siamese network as our network architecture, which is a particular neural network architecture consisting of two identical sub-networks with shared parameters, often used in a semi-supervised learning. The 'Siamese' networks share parameters and receive different inputs but are joined by an energy function layer, which computes some metrics between the highest-level feature representations, shown as Fig. 1.

We use a single layer CNN architecture proposed by Kim [17] as the basis network, which consists of a single convolutional layer, a max pooling layer and a fully connected hidden layer. We also add dropout after the fully connected layer to prevent overfitting.

The CNNs are optimized for pairs of text matrices and represent the inputs by a nonlinear mapping. The matrices are propagated through two same CNNs to be re-encoded as vectors with lower dimensions, and then compute the L_2 norm, a standard distance metric. The goal is to learn a discriminative representation in order to allow further specialization and generalization for future classifier training.

We try to design the loss function to minimize distance (x_i, x_j) when x_i and x_j are with the same labels, and maximize distance (x_i, x_j) when they have different labels. Using the L_2 norm as a distance metric, the *hinge* loss function is used as a regularization term:

$$h\left(x_i - x_j\right) = \begin{cases} \left\|x_i - x_j\right\|^2, & l_i = l_j \\ \max\left(0, m - \left\|x_i - x_j\right\|^2\right) & l_i \neq l_j \end{cases} \tag{1}$$

where l_i is the label of the text vector x_i, which has been re-encoded by the CNN hidden layer. The *hinge* loss is a kind of margin-based loss, which encourages those vectors with same labels to be close, and those dissimilar vectors to have a distance of at least m from each other. Because the *hinge* loss will result in expensive computing cost with high dimensional text data, we use the *Rmsprop* to optimize parameters.

The twin networks each have the following architecture: one convolution layer with 100 filters and three channels which is corresponding to three different filter size 3 × 300,4 × 300 and 5 × 300. Then a max-pooling layer and a fully connected layer are attached. The activation function is *ReLU*. Each sentence will be transformed as a 64 × 300 matrix, where one row is one word vector and all of sentences will be padded as length 64. Specially the filter width is equal to the width of word vectors, which means the convolution layer will reduce its input 64 × 300 matrix to (64-h + 1) × 1 feature map vector, where h is the filter height. Subsequently, the max-pooling layer will compress a feature map vector to a single max value. The fully connected layer then concentrate all of features into one vector. Then, the hinge loss between the two vector of the twin networks is used to learn a discriminative representation. Finally, the output is squashed into [0, 1] with a sigmoid function. We use the target 1 when the two sentences have the same class, and target 0 for a different class.

4 Network training

In our framework, there are two training stages: 1) pre-train a CNN model, 2) construct a Siamese CNN by the pre-trained CNN model, and fine-tune the Siamese CNN by few-shot learning strategy.

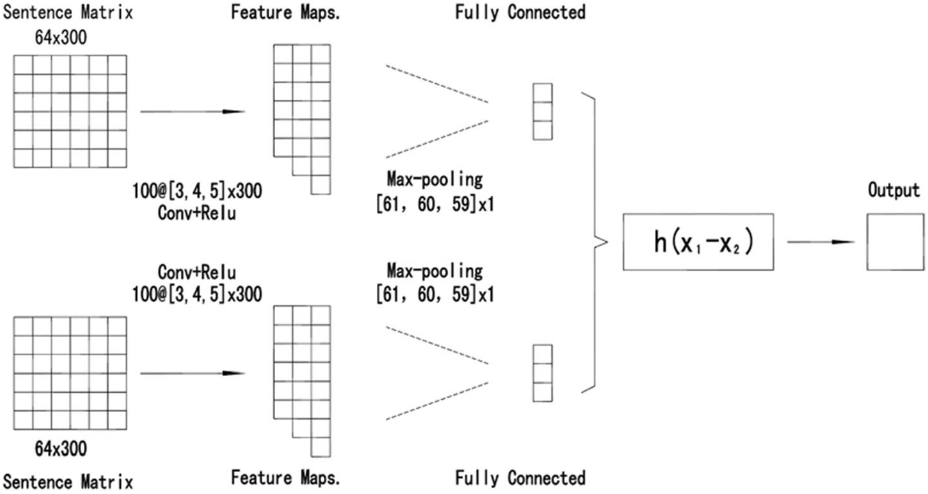


Fig. 1 A Siamese network architecture

4.1 Pre-train a CNN

Since training a CNN model is expensive in time consumption, we pre-train our CNN model to initialize the CNN parameters. The pre-trained CNN will transform the short text matrices to vectors with same length. The following is a detailed description.

At the beginning of network pre-training, the sentences should be tokenized and converted into word vectors by word embedding technology. Here we use word2vec. A tokenized sentence can be represented as $\mathcal{X} =< x_1, x_2, \cdots, x_i, \cdots x_n >$, where $x_i \in \mathcal{R}^k$, is a k-dimensional word vector, and all of sentences will be padded to length n. A slide window (filter) with d width will be applied to $\mathcal{X}$ to produce a subset $\mathcal{X}[i : i + d - 1]$. Here, we set the d as same as the length of word vector so as to keep the complete information in a word vector.

Let $\mathrm{W} \in \mathcal{R}^{dk}$ be a filter matrix, a feature map is defined as

$$o_i = f(W \circ \mathcal{X}[i : i + d - 1] + b), \tag{2}$$

where b is a bias term, $\circ$ is convolution operator. We choose the ReLU function $f(x) = max(0, x)$ as the activation function of the convolutional layer. When the filter is applied to each slide window in sentence $\mathcal{X}$, a sentence matrix is produced:

$$\mathcal{M} = [m_1, m_2, \cdots, m_i, \cdots m_{n-d+1}] \tag{3}$$

After convolutional layer, we then apply a max pooling operation to take the maximum value. The idea is that the most important features should be with the highest value. The fully connected layer concentrates all of feature mappings into a vector. Subsequently, a softmax classifier layer is connected to the CNN in order to back-propagate the re-build errors by supervised training strategy. In pre-training stage, we do not use fine-tuning to optimize the softmax classifier, because the dataset is imbalanced normally and the softmax classifier tend to identify all of samples as the major class.

In our experiments, the batch size is 50, dropout is set as 0.5, and the learning rate is 0.95. We use 100 filters in the convolution layer, and three channel 100 × 3 × 300, 100 × 4 × 300 and 100 × 5 × 300. The networks are trained 25 epoches to get optimized results.

4.2 Few-shot learning

For few-shot learning, we performed experiments by ‘episodes’. An ‘episode’ consists of a training set and a test set. The ‘episodes’ training strategy makes the classifiers more faithful to the test dataset and thereby improves generalization [25, 36]. A straightforward method to construct episodes is to choose *n* support samples for per class in order to match the expected situation and label unseen samples in test phase. However, the situation in short text classification is very different with image domain, where the text training set normally has a few class, such as positive, negative and neural, and there are hundreds of support samples for each class. Moreover, there are many different topics, phrases and sentence structures in each class.

In our experiments, we apply the idea of ‘prototypes’ in literature [29]. We view those sentences with different topics and typical sentence structure are different ‘prototypes’ in despite of whether their class labels are same or different. Besides, those samples that the pre-trained softmax classifier fails to identify are also put into the prototype set. Then, based on stochastic sampling strategy, each prototype sample will combine with *K* samples. The dataset can be represented as $\{[(x_1, x_2), d_1,], \ldots\ldots, [[(x_k, x_m), d_i],\}$, where (x_k, x_m) is a sample pair, and d_i is their label. If (x_k, x_m) have same class, then $d_i = 1$, otherwise, $d_i = 1$. The siamese network should be given a 1:1 ratio of same-class and different-class pairs to train on.

When the Siamese CNNs are trained by few-shot learning, a cost sensitive SVM classifier is connected to the network and replace the softmax classifier. Because the benchmark date are im-balanced data, we train the cost sensitive SVM classifier so as to identify the test data more precisely.

5 Experiments

In this section, we evaluate the performance of the method we proposed. Our experimental environment is: one Server which have 6 Intel CPUs and 2 kernels each CPU, 256 GB memory, one NVIDIA Tesla K20C GPU, Ubuntu 14.0; the code implementation is based on Keras and Theano. We take use of a pre-trained word2vec Google News model, the GoogleNews-vectors-negative300-SLIM.bin not GoogleNews-vectors-negative300.bin. The former can save more memory and include about 300 k English words, which is enough for our benchmark datasets.

The parameters of the Siamese CNN are set as: batch size is 50, dropout is set as 0.5, and the learning rate is 0.95. We use 100 filters in the convolution layer, and three channel 100 × 3 × 300, 100 × 4 × 300 and 100 × 5 × 300. The networks are trained 25 epoches. The pairing train data are divided into three parts. Then the final accuracy is taken as the mean accuracy of training on the three datasets.

5.1 Datasets

We choose three widely used twitter benchmark datasets and one game twitter dataset.

1) Multigames data

The Multigames was constructed by searching topics of games and downloaded from twitter. There are 12,780 tweets in this dataset, and all of them were annotated by hands, including positive 3952, neutral 7913 and negative 915. This dataset is produced originally by Prof Huajie Zhang's research group [40].

2) Hcr data

The Health Care Reform (HCR) dataset was created by crawling tweets. The original dataset have five sentiment labels: positive, negative, neutral, irrelevant and unsure. All the tweets were manually labeled by authors [31]. For our experiments, we discarded the irrelevant and unsure tweets. The dataset consists of 2394 tweets (470 neutral, 1382 negative, 542 positive).

3) SS-Tweet data

Sentiment Strength Twitter Dataset (SS-Tweet) was manually annotated which contains 4242 tweets. As seen in its name, the sentiments of tweets in this dataset have positive and negative sentiment strengths. The dataset was constructed by Thelwall et al. [22] to evaluate SentiStrenth. Saif et al. [27] proposed re-annotating tweets of this dataset with negative (a number 2), neutral (a number 0) and positive (a number 1), instead of sentiment strengths. The final dataset we used consists of 1037 tweets with a number 2 (negative), 1953 tweets with a number 0 (neutral) and 1252 tweets with a number 1(positive).

4) Semeval_b data

The SemEval-2013 Dataset (SemEval b) was constructed for the Twitter sentiment analysis task (Task 2) in the Semantic Evaluation of Systems challenge (SemEval-2013) [24]. The original SemEval dataset consists of 20 K tweets, and were all annotated by hands with three sentiment polarity labels: positive, neutral and negative. To collect data for the experiments, we tried to retrieve7967 tweets.

Normally, we take positive and negative samples to train and evaluate different methods. The sample numbers of twitter data we used are described in Table 1.

5.2 Comparing with traditional methods

At first, we compare our approach with some traditional machine learning methods, such as Adaboost(Ada), linear Support Vector Machine (SVM), Max Entropy, Random Forest(RF) and Naïve Bayes Network (NB). Some popular and widely used BOW text features are extracted from these benchmark twitter datasets for these methods, such as stop words removal, Sentiment WordNet (SentiWordNet), part-of-speech tagging (POS), Emoticon and Unigram. Considering the imbalanced distribution of different sentiment samples in dataset, some cost-

Table 1 the number of tweet samples

	Positive	Negative	Neural	Total
Multigames	3952	915	7913	12,780
Hcr	542	1382	470	2394
SS-Tweet	1252	1037	1953	4242
Semeval_b	2964	1151	3852	7967

sensitive SVMs [1, 2] classifiers would be more suitable choice than linear SVM. So, we choose the cost-sensitive Linear SVM as our classifier when implementing. Besides, we use random down-sampling method to weaken the imbalance when training models. According to the results described in the Table 2, the Random Forest and Linear SVM show better performance than other traditional classifiers, such as NB and Max Entropy. The Siamese CNN approach we proposed provides the most outstanding performance: outweighs NB, Max-Entropy, Adaboost, Random Forest and linear SVM on all four benchmark datasets in accuracy.

5.3 Comparing with deep learning methods

Moreover, we also compare our approach with three types of deep learning methods, the Simple RNN [23], LSTM [32] and DSC [40] which is based on autoencoders. The results are presented in Table 3.

The Simple RNN (SRNN) has the basic structure of RNN models, which have been a very competitive tool for Natural Language Processing. A Simple RNN has three basic layers: input layer, hidden layer and output layer. Long short-term memory (LSTM) models are a special kind of recurrent neural network, which has additional factors, like gates in electric circuit. LSTM has been widely used for text recognition, speech recognition, and time series prediction problems, etc. We append a *softmax* layer to SRNN and LSTM network as a classifier in our experiments. These two methods still use BoW features not word embedding features for the implementation reasons.

The RNN and LSTM is trained using *rmsprop* optimizer, which works better than annealed optimizer in our experiment. Activation function is set as *tanh*, which performs better than *ReLU*. Dropout layer is necessary as a regularizer, its rate is set 0.5. The hidden layer dimension is 200. The optimal performance is obtained after 10–15 training epoches. According to Table 3, the LSTM easily attains higher accuracy than simple RNN. Overall, the deep network methods, such as Simple RNN, LSTM, and DSC do not outweigh Siamese-CNN method in our experiments.

The imbalance in data is the main reason for those methods who get lower performance, especially for the traditional methods, such as Max Entropy, Naïve Bayes Network (NB), etc. The possible solution for this problem is to balance the distribution of different class samples in dataset. However, a large number of labelled samples are needed thus which is impossible in most of situations or the cost is too expensive to afford. It seems that our method does not be affected by the imbalance in data. The pairing training data would help eliminate the negative influence of imbalance to some extent.

Besides, the word2vec can help the CNN network get more valuable information. As we know, the sentence BoW vectors usually are so high dimensional and sparse, which means the classifiers need a very large amount of training samples. The word2vec can supply dense and continuous vector space for classifiers. Therefore, those methods with word embedding vector space can achieve more better performance.

Table 2 Performance of different traditional methods

	Ada	NB	Max En	RF	SVM
Multigames	75.0%	74.4%	75.6%	78.2%	77.2%
Hcr	64%	3.7%	57.0%	64.2%	62.6%
SS-Tweet	58.6%	59.9%	56.5%	58.8%	58.8%
Semeval_b	61.3%	60.5%	62.3%	64.9%	66.4%

Table 3 Accuracies of three deep learning methods

	SRNN	LSTM	DSC	Siamese-CNN
Multigames	66.7%	70.7	78.4	85.9%
Hcr	44.7%	54.5	64.0%	84.5%
SS-Tweet	41.4%	56.6	63.2%	89.1%
Semeval_b	47.6%	47.3	66.4	89.0%

6 Conclusion

In our study, we propose an efficient few-shot learning framework for short text classification. We use a pre-trained CNN to construct a Siamese network, and then apply few-shot learning strategy to improve. In order to maintain a gap between text vectors with different class labels and get much generalization, we introduce hinge loss function into CNN reconstruction cost. When using few-shot learning, we achieve improvement in both of accuracy and F1 score on multi-class twitter classification, and outweigh some traditional machine learning methods and a few deep learning approaches.

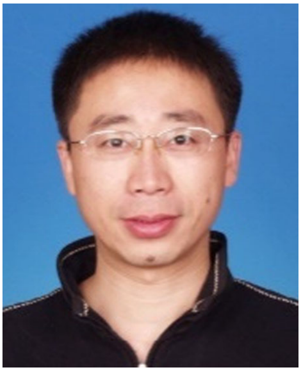
Acknowledgements This work is supported by the Chinese National Natural Science Foundation (NSFC) [grant numbers 61772281, 61602254]; the National Social Science Foundation of China (No. 16ZDA054); Jiangsu Provincial 333 Project (BRA2017396); Six Major Talents PeakProject of Jiangsu Province (XYDXXJS-CXTD-005); the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

References

1. Bin G, Sheng VS (2016) A robust regularization path algorithm for ν-support vector classification. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2016.2527796
2. Bin G, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. IEEE Trans Neural Netw Learn Syst 26(7):1403–1416
3. Blaes S, Burwick T (2017) Few-shot learning in deep networks through global prototyping[J]. Neural Netw Off J Int Neural Netw Soc 94:159–172
4. Chen B, Qi X, Sun X, Shi Y-Q (2017) Quaternion pseudo-Zernike moments combining both of RGB information and depth information for color image splicing detection. J Vis Commun Image Represent
5. Cheng J, Zhang X, Li P et al (2016) Exploring sentiment parsing of microblogging texts for opinion polling on Chinese public figures. Appl Intell 45(2):429–442
6. Ding G, Guo Y, Zhou J, Gao Y (2016) Large-scale cross-modality search via collective matrix factorization hashing. IEEE Trans Image Process 25(11):5427–5440
7. Ding G, Zhou J, Guo Y, Lin Z, Zhao S (2017) Large-scale image retrieval with sparse embedded hashing. Neurocomputing 257:24–36
8. Fu Z, Huang F, Sun X, Vasilakos AV, Yang C-N (2016) Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2016.2622697
9. Guo Y, Ding G, Han J (2017) Robust quantization for general similarity search. IEEE Trans Image Process PP(99):1–1
10. Guo Y, Ding G, Liu L, Han J, Shao L (2017) Learning to hash with optimized anchor embedding for scalable retrieval. IEEE Trans Image Process 26(3):1344–1354
11. Guo Y, Ding G, Han J et al Zero-shot learning with transferred samples. IEEE Trans Image Process 26(7):3277
12. Han J, Cheng G, Li Z et al (2017) A unified metric learning-based framework for co-saliency detection. IEEE Trans Circuits Syst Video Technol PP(99):1–1
13. Han J, Chen H, Liu N et al (2017) CNNs-based RGB-D saliency detection via cross-view transfer and multiview fusion[J]. IEEE Trans Cybern PP(99):1–13
14. Hariharan B, Girshick R (2016). Low-shot visual object recognition. arXiv:1606.02819
15. Hecht T, Gepperth A (2016). Computational advantages of deep prototype-based learning. In: International conference on artificial neural networks, Springer, pp 121–127

16. Jetley S, Romera-Paredes B, Jayasumana S, Torr P (2015) Prototypical priors: from improving classification to zero-shot learning. arXiv preprint arXiv:1512. 01192
17. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv, 1408.5882
18. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition. Proceedings of the 32nd international conference on machine learning, Lille, France
19. Lake BM, Salakhutdinov R, Tenenbaum JB (2013) One-shot learning by inverting a compositional causal process[J]. Adv Neural Inf Proces Syst 2526–2534
20. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: IEEE conference on computer vision and pattern recognition. CVPR 2009 IEEE, pp 951–958
21. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. IEEE Trans Inf Forensics Secur 10(3):507–518
22. Mike T, Kevan B, Georgios P (2012) Sentiment strength detection for the social web. J Assoc Inf Sci Technol 63(1):163–173
23. Mikolov T, Karafiát M, Burget L et al (2010) Recurrent neural network based language model. 11th Annual Conference of the International Speech Communication Association, Makuhari, Japan, pp 1045–1048
24. Nakov P, Rosenthal S, Kiritchenko S et al (2016) Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. Lang Resour Eval 50(1):35–65
25. Ravi S, Larochelle H (2017) Optimization as a Model for Few-Shot Learning. 5th International Conference on Learning Representations(ICLR), Toulon, France. https://openreview.net/pdf?id=rJY0-Kcll
26. Rezende DJ, Mohamed S, Danihelka I, Gregor K, Wierstra D (2016) One-shot generalization in deep generative models. arXiv preprint arXiv:1603.05106
27. Saif H, Fernández M, He Y et al (2013) Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the STS-gold. Proceedings of the first international workshop on emotion and sentiment in social and expressive media: approaches and perspectives from AI, A workshop of the XIII International Conference of the Italian Association for Artificial Intelligence, Turin, Italy, pp 9–21
28. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
29. Snell J, Swersky K, Zemel RS (2017) Prototypical networks for few-shot learning. arXiv:1703.05175
30. Socher R, Lin CC-Y, Ng AY, Manning CD (2011) Parsing natural scenes and natural language with recursive neural networks. Proceedings of the 28th international conference on machine learning, Washington, USA, pp 129–136
31. Speriosu M, Upadhyay S, Sudan N et al (2011) Twitter polarity classification with label propagation over lexical links and the follower graph. Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP, Edinburgh, Scotland, pp 53–63
32. Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. 13th annual conference of the international speech communication association, Portland, USA, pp 194–197
33. Tang D, Wei F, Qin B (2014) Coooolll: A deep learning system for Twitter sentiment classification. Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, pp 208–212
34. Triantafillou E, Zemel RS, Urtasun R Few-shot learning through an information retrieval lens. arXiv: 1707.02610
35. Turney PD, Pantel P (2010) From frequency to meaning: vector space models of semantics. J Artif Intell Res 37(1):141–188
36. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. Adv Neural Inf Process Sys 3630–3638
37. Wang X, Liu Y, Sun C et al (2012) Predicting polarities of tweets by composing word embeddings with long short-term memory. Unabbreviated Name of Conference, Portland, USA, pp 194–197
38. Wang J, Li T, Shi Y-Q, Lian S, Ye J Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics. Multimed Tools Appl. https://doi.org/10.1007/s11042-016-4153-0
39. Weinberger KQ, Blitzer J, Saul LK (2005) Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems, pp 1473–1480
40. Yan L, Zheng W, Zhang H(H) et al (2017) Learning discriminative sentiment chunk vectors for twitter sentiment analysis. J Inf Technol 18(7):1605–1613. https://doi.org/10.6138/JIT.2017.18.7.20170410
41. Yao X, Han J, Cheng G, Qian X, Guo L (2016) Semantic annotation of high-resolution satellite images via weakly supervised learning. IEEE Trans Geosci Remote Sens 54(6):3660–3671
42. Yao X, Han J, Zhang D, Nie F (2017) Revisiting co-saliency detection: a novel approach based on two-stage multi-view spectral rotation co-clustering. IEEE Trans Image Process 26(7):3196–3209
43. Zhang Z, Saligrama V (2015) Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE international conference on computer vision, pp 4166–4174
44. Zhang D, Han J, Li C, Wang J, Li X (2016) Detection of co-salient objects by looking deep and wide. Int J Comput Vis 20(2):215–232

45. Zhang D, Han J, Jiang L, Ye S, Chang X (2017) Revealing event saliency in unconstrained video collection. IEEE Trans Image Process 26(4):1746–1758
46. Zhang D, Meng D, Han J (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39(5):865–878
47. Zhao Y, Ding DZ, Chen RS (2016) A discontinuous Galerkin time domain integral equation method for electromagnetic scattering from PEC objects. IEEE Trans Antennas Propag 64(6):2410–2417
48. Zheng Y, Jeon B, Sun L, Zhang J, Zhang H (2017) Student's t-Hidden Markov Model for Unsupervised Learning Using Localized Feature Selection. IEEE Trans Circuits Syst Video Technol. https://doi.org/10.1109/TCSVT.2017.2724940
49. Zhou Z, Yang C-N, Chen B, Sun X, Liu Q, Wu QMJ (2016) Effective and efficient image copy detection with resistance to arbitrary rotation. IEICE Trans Inf Syst E99-D(6):1531–1540
50. Zhou Z, Wang Y, Jonathan Wu QM, Yang C-N, Sun X (2017) Effective and efficient global context verification for image copy detection. IEEE Trans Inf Forensics Secur 12(1):48–63



Leiming Yan received his PhD in computer science from School of Computer Science and Engineering, Southeast University, China, in 2010. He is currently a lecturer in School of Computer and Software, Nanjing University of Information Science and Technology, China. He was a visiting scholar of Faculty of Computer Science, University of New Brunswick, Canada, from May 2015 to May 2016. His research interests include big data mining, deep learning and complex networks.

Yuhui Zheng received the Ph.D. degree in the Nanjing University of Science and Technology, China in 2009. Now, he is an associate professor in the School of Computer and Software, Nanjing University of Information Science and technology. His research interests cover image processing, pattern recognition, and remote sensing information system.

Jie Cao received the Ph. D. degree in Management Science and Engineering from Southeast University, Nanjing, China in 2005. He is currently a Professor with the School of Mathematical &Statistics, Nanjing University of Information Science and Technology, China. His research interests include Complex System Analysis and Management Decision, Emergency Management, Information Management and Information System, Financial Engineering.