PAPER

An Active Transfer Learning Framework for Protein-Protein Interaction Extraction

Lishuang LI[†a)], Xinyu HE[†b)], Jieqiong ZHENG[†c)], Degen HUANG[†d)], *Nonmembers*, *and* Fuji REN[††], *Member*

SUMMARY Protein-Protein Interaction Extraction (PPIE) from biomedical literatures is an important task in biomedical text mining and has achieved great success on public datasets. However, in real-world applications, the existing PPI extraction methods are limited to label effort. Therefore, transfer learning method is applied to reduce the cost of manual labeling. Current transfer learning methods suffer from negative transfer and lower performance. To tackle this problem, an improved TrAdaBoost algorithm is proposed, that is, relative distribution is introduced to initialize the weights of TrAdaBoost to overcome the negative transfer caused by domain differences. To make further improvement on the performance of transfer learning, an approach combining active learning with the improved TrAdaBoost is presented. The experimental results on publicly available PPI corpora show that our method outperforms TrAdaBoost and SVM when the labeled data is insufficient,and on document classification corpora, it also illustrates that the proposed approaches can achieve better performance than TrAdaBoost and TPTSVM in final, which verifies the effectiveness of our methods.

key words: *protein-protein interaction, TrAdaBoost, actively transfer learning, relative distribution*

1. Introduction

With the rapid development of information digitalization and biomedicine, biomedical literatures are expanding at an exponential rate, which makes manually detecting the required information difficult. As one of the most important biomedical text mining branch, Protein-Protein Interactions Extraction (PPIE) plays an important role in establishing protein knowledge network and constructing ontology.

Current methods for PPIE fall into three main categories: word co-occurrence, pattern matching and statistical machine learning [1], [2]. Compared with other methods, machine learning methods are more robust. So far there have been many attempts to develop machine learning techniques to extract protein-protein interaction pairs. These techniques include feature vectors-based, kernel-based [1] and combination methods [2]. For example, Zhang [1] presented a weighted multiple kernels learning-based approach, which included feature-based, tree, graph and POS path kernels and achieved 64.41% F-score on AIMed, 65.84% F-score on BioInfer, 74.38% F-score on HPRD50, 75.73% F-score on IEPA and 83.01% F-score on LLL. Li [2] combined feature-based kernel, tree kernel with semantic kernel which obtained an F-score of 69.40% on AIMed. In recent years, some researchers have utilized word embeddings and deep learning for PPIE, For example, Li [3] proposed an approach capturing external information from the web-based data which involved distributed representation, vector clustering and Brown clusters word representation techniques and achieved F-scores of 69.0%, 74.1%, 78.0%, 76.3% and 87.8% on AIMed, BioInfer, HPRD50, IEPA and LLL respectively.

Although many PPI extraction systems have achieved good results on five public datasets, i.e. Aimed, BioInfer, HPRD50, IEPA and LLL, most of them were evaluated on the test corpus which has the same distributions with the training corpus, and thus when other domains data are tested, the performance will greatly decrease due to the distribution change. For example, the result is usually not satisfied when the PPI extraction model trained from drosophila data is evaluated on human data, and since no single corpus is large enough to saturate data of all species, which often requires large and expensive cost. Therefore, how to make full use of the knowledge of other domain with similar data distribution is still a challenge.

One of the current methods to alleviate the above problem is transfer learning, which has been obtained better results in web text data mining [4], document classification [5], [6]. The main idea of transfer learning is to utilize the knowledge from other domain(s) to help learn the current domain. For example, to improve the performance of PPI extraction on different distribution corpora, Miwa [7] used one of the entire corpora as the target corpus and adjusted the weights of the remaining corpora (source corpora) with inductive transfer learning method SVM_CW on five public biomedical corpora. The experiment was conducted on the target corpus which is still large and the situation that target data deficiency was not analyzed in this paper. In biomedical field, the labeling cost by experts is huge, therefore, it is necessary to improve the performance of the machine learning method when the target data are not enough.

To solve the low performance problem of machine learning when target corpus is insufficient, Dai [8] decreased the negative effects of source domain and boosted the accuracy on the target domain by Boosting, and achieved better performance on the document classification corpora. However, the knowledge transferred from other domains may re-

Manuscript received July 15, 2017.
Manuscript publicized October 30, 2017.

[†] The authors are with the Dalian University of Technology, Dalian, China.
[††] The author is with the University of Tokushima, Tokushima-shi, 770–8501 Japan.
a) E-mail: lilishuang314@163.com (Corresponding author)
b) E-mail: hexinyu@mail.dlut.edu.cn
c) E-mail: zjq2014@mail.dlut.edu.cn
d) E-mail: huangdg@dlut.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2017EDP7232

duce the learning accuracy due to implicit domain differences, this phenomenon is called negative transfer which is one of the main problems in this area. Despite the fact that how to avoid negative transfer is a very important issue, little research work has been published on this topic. Rosenstein et al. [9] empirically showed that if two tasks were too dissimilar, then brute-force transfer might hurt the performance of the target task. Some works have been exploited to analyze relatedness among tasks and task clustering techniques, such as [10], they propose a data generating mechanism, which might help provide guidance on how to avoid negative transfer automatically.

Active learning is another approach to solve the label effort problem. Its basic idea is to select optimal samples by reducing the expected error of the learner, which mainly focuses on selecting a few suitable examples to label by experts. Thus, different active learners have different selection criteria. For example, uncertainty sampling is the simplest measure to select the example on which the current learner has lower certainty [11]. The Query-by-Committee method selects the examples that cause maximal disagreement to achieve better accuracy [12]. However, the obvious issue in this area is the cost associated with the answer from domain experts.

In this paper, to reduce the negative transfer and improve the accuracy, we propose a new actively transfer framework ActTrAdaBoost to extract PPIs. Firstly, an improved TrAdaboost algorithm (RDTrAdaboost) is proposed to avoid negative transfer by adjusting the weights of the source datasets instances. In RDTrAdaboost algorithm, the relative distribution [13] is introduced to avoid negative impacts caused by domain difference. Secondly, we combine RDTrAdaboost algorithm with active learning to improve the performance by adding data labeled by experts. The results show that our method performs better than TrAdaBoost on PPI public corpora.

2. Methodology

A workflow to describe our method is shown in Fig. 1. The input of our method contains three annotated corpora: AIMed, HPRD50 and IEPA. Pre-processing is operated using the OpenNLP tools. RDTrAdaboost is the improved TrAdaboost, which will be illustrated in Sect. "2.2 Improved TrAdaBoost". Feature extractor will be described in Sect. "2.1 Feature extractor" and Sect. "2.3 Actively transfer learning" will describe more details of ActTrAdaBoost. The output is the classification results of protein pairs, treating the positive instances as the interacting pairs while the negative instances as the non-interacting pairs.

2.1 Feature Extractor

We select five features as our baseline features and they are described as follows:

Words from protein names: all the words in two protein names are included.

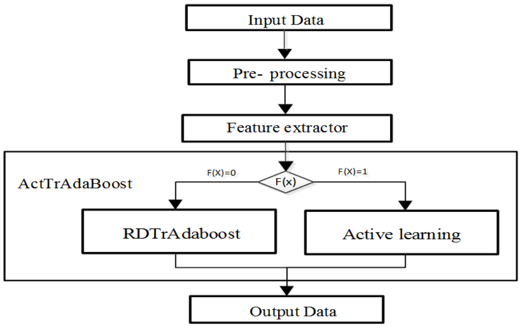


Fig. 1 Workflow ofActTrAdaBoost

Words surrounding two protein names: these features include n words on the left side of the first protein name and n words on the right side of the second protein name respectively. n is set to 5 in our experiments. If there is no word surrounding two protein names, "NULL" will be used.

Words between two protein names: these features include all words that are located between two protein names. If no word appears between two protein names, the feature will be "NULL".

Interaction term: a sentence is considered containing PPI information only if it includes at least one interaction word or keyword (such as "regulate", "interact", "modulate", etc.). If there is one keyword between or among the surrounding words of two protein names, the keyword is set as the keyword feature; if there is no keyword, the keyword feature will be set to "NULL".

Distance feature: from the corpus, we find that the protein pair is more likely to have interaction relation if the distance (the number of words) between the two proteins is short. The distance feature can be divided into two classes:

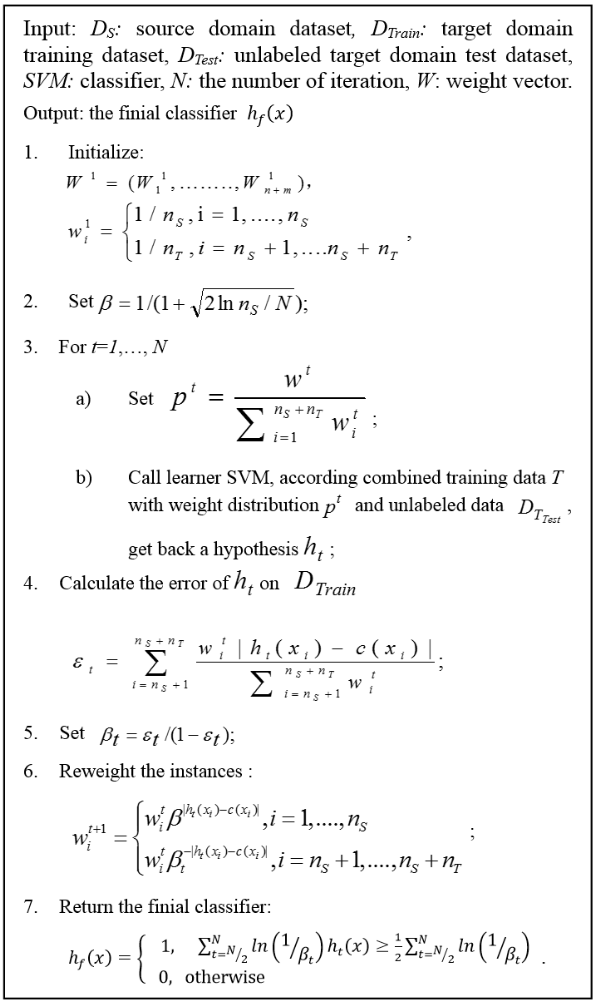
- The number of the non-proteins words between two proteins (Word-Num).
  If Word-Num≤3, the feature value will be set to "1"; if 3<Word-Num≤6, it will be set to "2"; if 6<Word-Num≤9, it will be set to "3"; else, it will be set to "4".
- The number of the protein names between two proteins. If no other proteins appear between the two proteins, the feature value will be set to "0"; otherwise, it is the number of other proteins.

2.2 Improved TrAdaBoost

In this section, our RDTrAdaBoost (Relative Distribution TrAdaBoost) algorithm is presented to improve TrAdaBoost. A formal description of TrAdaBoost is given in Fig. 2.

Some definitions are given as follows :

Definition 2.1 (symbols):
X: input space; Y = {-1, +1}: output space; $c(x)$: the label of x, and $c(x)\in Y$.



```
Input: $D_S$: source domain dataset, $D_{Train}$: target domain training dataset, $D_{Test}$: unlabeled target domain test dataset, $SVM$: classifier, $N$: the number of iteration, $W$: weight vector.
Output: the finial classifier $h_f(x)$
1.  Initialize:
    $W^1 = (W_1^1, \ldots\ldots, W_{n+m}^1)$,
    $w_i^1 = \begin{cases} 1/n_S, \mathrm{i} = 1, \ldots, n_S \\ 1/n_T, i = n_S + 1, \ldots n_S + n_T \end{cases}$,
2.  Set $\beta = 1/(1+\sqrt{2\ln n_S / N})$;
3.  For $t=1,\ldots,N$
    a)  Set $p^t = \frac{w^t}{\sum_{i=1}^{n_S+n_T} w_i^t}$;
    b)  Call learner SVM, according combined training data $T$
        with weight distribution $p^t$ and unlabeled data $D_{T_{Test}}$,
        get back a hypothesis $h_t$;
4.  Calculate the error of $h_t$ on $D_{Train}$
    $\varepsilon_t = \sum_{i=n_S+1}^{n_S+n_T} \frac{w_i^t \, | h_t(x_i) - c(x_i) |}{\sum_{i=n_S+1}^{n_S+n_T} w_i^t}$;
5.  Set $\beta_t = \varepsilon_t / (1-\varepsilon_t)$;
6.  Reweight the instances :
    $w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i)-c(x_i)|}, i = 1, \ldots, n_S \\ w_i^t \beta_t^{-|h_t(x_i)-c(x_i)|}, i = n_S+1, \ldots, n_S+n_T \end{cases}$;
7.  Return the finial classifier:
    $h_f(x) = \begin{cases} 1, & \sum_{t=N/2}^{N} ln\left(1/\beta_t\right) h_t(x) \geq \frac{1}{2}\sum_{t=N/2}^{N} ln\left(1/\beta_t\right) \\ 0, & \text{otherwise} \end{cases}$.
```

Fig. 2 Algorithm description of TrAdaBoost

Definition 2.2 (Dataset):

$$D_S = \{x_{s_i}, y_{s,i}\}_{i=1}^{n_s}, D_{T_{Test}} = \{x_{T_i}\}_{i=1}^{n_{Test}}, D_{T_{Train}} = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_{Train}},$$
$$D_T = \{D_{T_{Test}}, D_{T_{Train}}\}, T = D'_S \cup_{D_{T_{Train}}} \textit{where } D'_S \subseteq D_S,$$

where D_S means the source dataset, D_T represents the target dataset, $D_{T_{Train}}$ is the target training dataset, $D_{T_{Test}}$ means the target test dataset, T refers to the combined training data, D'_S means the weighted sampling of D_S, n_S and n_T represent the size of the source dataset and the target dataset respectively, n_{Train} and n_{Test} represent the size of the target training dataset and target test dataset respectively and y_{S_i},y_{T_i} ∈{-1,+1}.

Leveraging that TrAdaBoost sets the source domain data with the same initial weight, our RDTrAdaBoost algorithm tries to initialize weights with relative distribution which is defined as (1),

$$\delta(x) = \frac{P_T(x)}{P_S(x)}, \tag{1}$$

where $P_T(x)$ and $P_S(x)$ represent the occurrence frequency of instance x in the target and source dataset respectively.

The rationality of our method is proved by the following theoretical analysis. The instance based transfer learning method has hypothesis as follows:

a) $P(Y_S|X_S)=P(Y_T|X_T)$

b) $X_S \approx X_T$

c) $P(X_S) \neq P(X_Y)$,

where $P(Y_S|X_S)$ and $P(Y_T|X_T)$ refer to the conditional distribution of the source and the target domain respectively. X_S and X_T represent the feature space of source dataset and target dataset respectively. $P(X_S)$ and $P(X_Y)$ represent the marginal distribution of source and target domain. The hypothesis demonstrates that when the source dataset and target dataset have the same conditional distribution, their marginal distribution is usually different. In this research, the distributions of source dataset and target dataset meet Eqs. (a)–(c).

The goal of transfer learning is to make the observed value from target dataset D_T close to the objective value. Therefore, we define a loss function $l(x,y,\theta)$, which is used to calculated the cost between the observed value and the objective value. In this function, x represents the input instances, y refers to the predicted result (0 or 1), θ is a variable parameter that is used to adjust the loss function l. To minimize the loss function, Zadrozny [14] made derivation of $l(x,y,\theta)$ as (2):

$$\begin{aligned} \theta^* &= argminE_{(x,y)\sim P_T}[l(x,y,\theta)] \\ &= argminE_{(x,y)\sim P_T}[\frac{P_S(x,y)}{P_S(x,y)} l(x,y,\theta)] \\ &= argmin \int_y \int_x P_T(x,y)\Big([\tfrac{P_S(x,y)}{P_S(x,y)} l(x,y,\theta)] \Big) dxdy \\ &= argmin \int_y \int_x P_S(x,y)\Big([\tfrac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta)] \Big) dxdy \\ &= argminE_{(x,y)\sim P_S}[\frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta)] \\ &= argminE_{(x,y)\sim P_S}[\frac{P_T(x)P_T(y|x)}{P_S(x)P_S(y|x)} l(x,y,\theta)] \\ &= argminE_{(x,y)\sim P_S}[\frac{P_T(x)}{P_S(x)} l(x,y,\theta)] \end{aligned} \tag{2}$$

To make the problem general, we introduce the penalty parameter. In addition, the $\frac{P_T(x)}{P_S(x)}$ in (2) is same to $\delta(x_i)$ according to (1), therefore, the optimization problem can be written as (3):

$$\theta^* = argmin \sum_{i=1}^{ns} \delta(x_i) l(x_{S_i}, y_{S_i}, \theta) + \lambda\Omega(\theta), \tag{3}$$

where λ is regularization coefficient, and $\Omega(\theta)$ is regularization term. As the value of loss function $l(x,y,\theta)$ is fixed, therefore, the purpose of Eq. (3) is solving the minimum of $\delta(x_i)$. Therefore, we can see that the improved loss functions in the target domain are smaller than the original, the same procedure may be easily adapted in the source domain:

$$minE_{(x,y)\sim P_T}[l(x,y,\theta)] \le E_{(x,y)\sim P_T}[l(x,y,\theta)] \tag{4}$$

$$minE_{(x,y)\sim P_S}[l(x,y,\theta)] \le E_{(x,y)\sim P_S}[l(x,y,\theta)] \tag{5}$$

In final, the initial weight vector of our RDTrAdaBoost is defined as (6), which replaces the original initial weight vector in Fig. 2:

$$W^1 = (W_1^1, \ldots\ldots, W_{n_S+n_T}^1),$$

$$w_i^1 = \begin{cases} \delta(x)/n_S, & i = 1,\ldots,n_S \\ 1/n_T, & i = n_S+1,\ldots,n_S+n_T \end{cases} \tag{6}$$

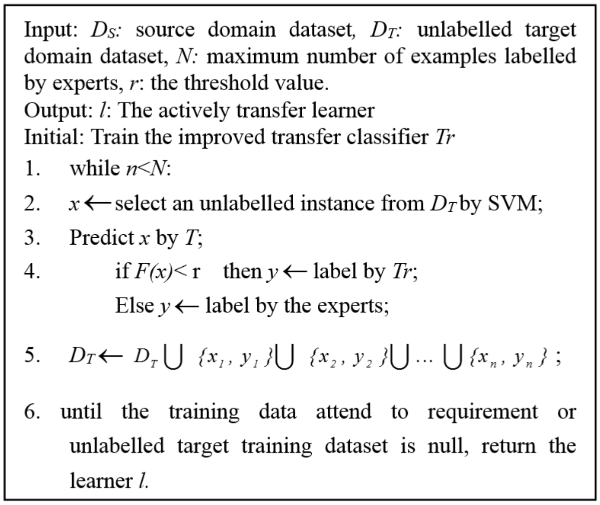
In Formula (6), n represents n_S, which means the size of the source dataset; m refers to n_T, which is the size of the target dataset. When "i=1,. . . .,n_S", the weight vector is expressed as "$w_i^1 = \delta(x)/n_S$"; When "i=n_S+1,. . . .,n_S+n_T", the weight vector is expressed as "$w_i^1 = 1/n_T$", $\delta(x)$ is defined as Formula (1), which means the relative distribution.

2.3 Actively Transfer Learning

Our actively transfer learning (ActTrAdaBoost) framework is introduced in this Section. Firstly, select an example which has the lowest confidence given by SVM. Secondly, predict the selected example x by the transfer classifier. Eventually, determine whether the example is labeled by experts or transfer classifier via decision function. Decision function $F(x)$ is defined as (7):

$$F(x) = [P(T_r(x) = y|x) > R] \bullet [L > N], \tag{7}$$

where T_r represents the improved transfer classifier (RDTrAdaBoost), $P(T_r$(x)=$y\big|x)$ is the classification confidence of x given by the transfer learning classifier, R refers to the threshold value of $P(T_r$(x)=$y\big|x)$ represents the size of training dataset, L represents the size of training dataset. N stands for the maximum number of examples labeled by experts. Formula (7) provides two conditions for the selection. We choose the active learning method when $P(T_r$(x)=$y\big|x)<R$ *or* $L<N$ is true, namely, if the classification confidence of the transfer classifier is low or the size of the target training dataset is small, the selected example should be labeled by the experts. Otherwise, the transfer classifier is selected. The main process of ActTrAdaBoost is summarized in Fig. 3.



```
Input: $D_S$: source domain dataset, $D_T$: unlabelled target domain dataset, $N$: maximum number of examples labelled by experts, $r$: the threshold value.
Output: $l$: The actively transfer learner
Initial: Train the improved transfer classifier $Tr$
1.  while $n<N$:
2.  $x \leftarrow$ select an unlabelled instance from $D_T$ by SVM;
3.  Predict $x$ by $T$;
4.          if $F(x)<$ r    then $y \leftarrow$ label by $Tr$;
            Else $y \leftarrow$ label by the experts;
5.  $D_T \leftarrow D_T \bigcup \{x_1, y_1\} \bigcup \{x_2, y_2\} \bigcup \cdots \bigcup \{x_n, y_n\}$;
6.  until the training data attend to requirement or
    unlabelled target training dataset is null, return the
    learner $l$.
```

Fig. 3 Algorithm description of actively transfer learning

3. Experiments

3.1 Data Set

Our method is evaluated on three public biomedical corpora: AIMed [15], HPRD50 [16] and IEPA [17]. AIMed is from Medline database, which has 1000 pairs of positive instances, 3500 pairs of negative instances. IEPA is from PubMed, which has 336 pairs of positive instances, 336 pairs of negative instances. HPRD50 has 163 pairs of positive instances, 270 pairs of negative instances. They are generally used in the assessment of PPIE methods with slightly different annotating policies.

3.2 Evaluation Measures

We evaluate our method by F-score, which is the harmonic mean of Precision and Recall. The definition of Precision (P), Recall (R) and F-score (F) are shown in (8), (9), (10) respectively, where TP is short for true positives, FP represents false positives, and FN stands for false negatives.

$$P = TP/(TP + FP) \tag{8}$$

$$R = TP/(TP + FN) \tag{9}$$

$$F - score = 2 * P * R/(P + R) \tag{10}$$

3.3 KL Distance

In order to characterize the distribution differences among the corpora, KL distance is introduced, which is defined as (11):

$$D(P||Q) = \sum_{x \in X} P(x) log \frac{P_(x)}{Q_(x)} \tag{11}$$

where *P(x)* is the word frequency that x appears in dataset *P*. Similarly, we can get Q(x), $D(P\big\|Q)$ stands for KL distance. Generally, KL distance is asymmetric because the distance from P to Q is usually not equal to Q to P. Table 1 shows the KL distances. The comparison of KL distance is described as (12):

Table 1 Results of keywords extraction

Theoretical distribution Q	Actual distribution P(AIMed)	Actual distribution P(HPRD50)	Actual distribution P(IEPA)
AIMed	0	6.94E6	7.08E6
HPRD50	1.15E7	0	1.45E7
IEPA	9.72E6	1.36E7	0

$$D(P_{IEPA}\|Q_{Aimed}) < D(P_{Aimed}\|Q_{Hprd50}) < D(P_{Aimed}\|Q_{IEPA}) < D(P_{IEPA}\|Q_{Hprd50}) \tag{12}$$

3.4 Experiment Setting

To verify the effectiveness of the presented method, we select a tiny amount of the target training data (20% of data from the training set). In addition, the target corpora should have different distributions with the source corpora while the domains should be similar. Therefore, four groups of PPIE experiments based on different target and source datasets are designed. In the experiments, IEPA and AIMed are selected as the target corpora and the rest as the source corpora respectively. All the results are measured by 5-fold cross-validation. The baseline employs SVM with a linear kernel. In the experiments, to simulate the environment of less training data mentioned in introduction, we randomly choose 20% data from the target dataset for training and the rest 80% for testing based on SVM and RDTrAdaBoost. As for ActTrAdaBoost, the 20% of data are set to be unlabeled initially, the decision function will determine how to label the unlabeled examples. The experimental purpose is to verify the effectiveness of our method solving labeled data deficient problems. Therefore, we focus on comparing the results when the proportion of examples labeled by experts in target dataset is 2% and the entire trend of the curves. Most related works with PPI adopted 5-fold (80% data is used for training) or 10-fold cross-validation (90% data is used for training). The objective of our experiment is to alleviate data insufficient problem which is different from other PPI literature, so these related works are not compared in experiments.

3.5 AIMed as Target Corpus

In this section, we compare our algorithms, i.e. the improved TrAdaBoost (RDTrAdaBoost) and ActTrAdaBoost, with the original TrAdaBoost and SVM when AIMed is selected as target domain.

Figure 4 shows that, when IEPA is selected as the source domain, the performances of TrAdaBoost, RDTrAdaBoost and ActTrAdaBoost are much better than SVM. It reveals that transfer learning is better than SVM on AIMed. Our method has no significant advantages compared with original TrAdaBoost on this corpus because the KL distance from IEPA to AIMed is small due to the lower domain differences.

When HPRD50 is selected as the source domain, we can observe from Fig. 5 that our RDTrAdaBoost and ActTrAdaBoost both outperform TrAdaBoost and SVM. The results suggest that initializing weights with relative distribution can contribute to accelerate the convergence. Eventually, three transfer learning methods all achieve reasonable results, which indicates that the negative transfer does not occur in the experiment. In addition, we compare our transfer learning methods with TrAdaBoost when data are insufficient, for example when the proportion of examples labeled by experts in target dataset is 2%, ActTrAdaBoost outperforms TrAdaBoost by 1.75% in Fig. 4, by 4.41% in Fig. 5.

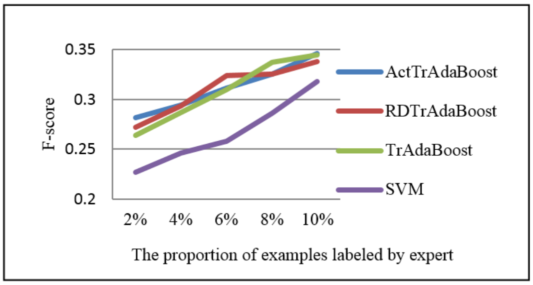


Fig. 4 IEPA as source domain, the F-scores of different methods on AIMed

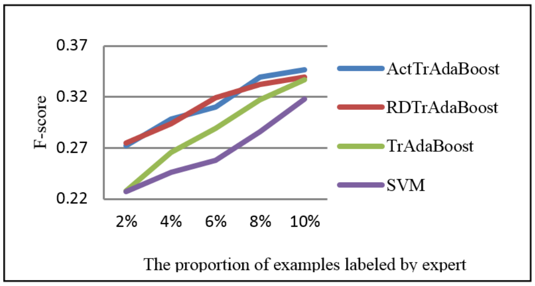


Fig. 5 HPRD50 as source domain, the F-scores of different methods on AIMed

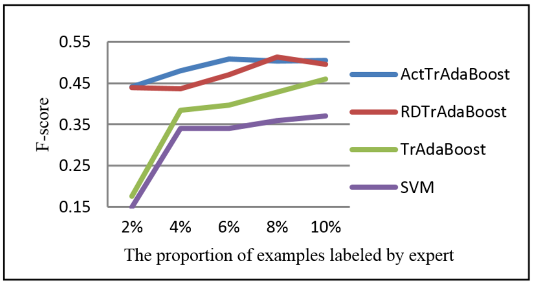


Fig. 6 AIMed as source domain, the F-scores of different methods on IEPA

3.6 IEPA as Target Corpus

Figure 6 and Fig. 7 compare the performance of our method with TrAdaBoost and SVM when IEPA is selected as target domain.

Figure 6 shows that, when IEPA is selected as the source domain, the performance of our RDTrAdaBoost and ActTrAdaBoost both outperform TrAdaBoost. Especially, ActTrAdaBoost outperforms TrAdaBoost by 26.69% F-score when the proportion of examples labeled by experts in target dataset is 2%. It can be concluded that the transfer performance can be greatly improved by adjusting the initial weights of TrAdaBoost.

In Fig. 7, the performance of TrAdaBoost is worse than SVM when the labeled proportion is 4%, which in-

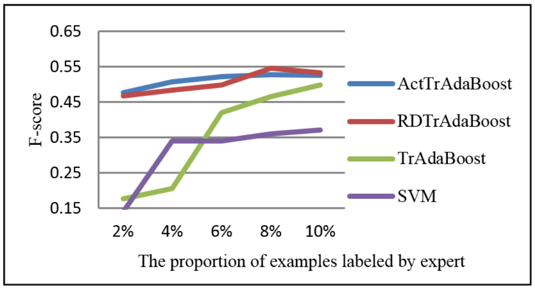


Fig. 7 HPRD50 as source domain, the F-scores of different methods on IEPA

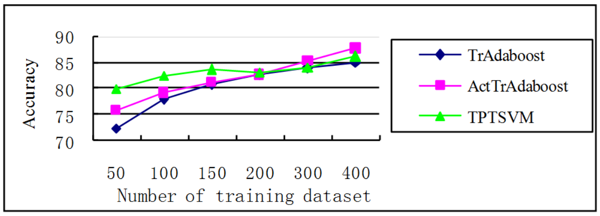


Fig. 8 The accuracy of different methods on document classification corpora

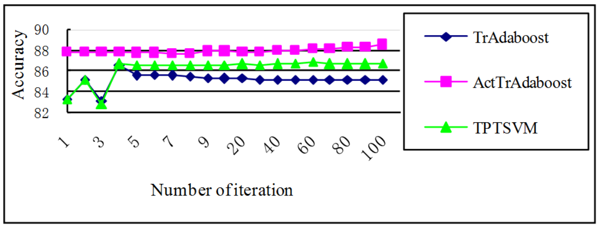


Fig. 9 Learning curves on document classification corpora

dicates that the negative transfer occurrs on IEPA. We can also see from the formula (12), the KL distance from IEPA to Hprd50 is the largest which means the higher risk of distribution differences. When the train data are insufficient (2%~6%), it can be observed that our RDTrAdaBoost and ActTrAdaBoost significantly outperform TrAdaBoost. Especially, ActTrAdaBoost outperforms TrAdaBoost by 29.9% when the proportion of examples labeled by experts in target dataset is 2%, which further illustrates our transfer learning method can reduce the impact of the data deficiency, avoiding the negative transfer problem.

Additionally, in Fig. 6 and Fig. 7, ActTrAdaBoost outperforms RDTrAdaBoost when the proportion of examples labeled by experts in target dataset is 4% and 6%, which shows that actively transfer learning is effective. It indicates that adding the examples labeled by experts is effective for improving the extraction accuracy. With the increase of labeled data, the advantage of ActTrAdaBoost and RDTrAdaBoost is not obvious, and the F-scores of ActTrAdaBoost are close to RDTrAdaBoost in final. In conclusion, RDTrAdaBoost effectively solves the negative transfer problem in transfer learning. ActTrAdaBoost can reduce the number of examples labeled by experts and improve the performance of transfer learning.

3.7 Comparisons between Our Method and Other Works

We also conduct experiments on document classification corpora and evaluate our method by Accuracy which is defined as (13):

$$Accuracy = \frac{True}{True + False} \tag{13}$$

Figure 8 shows the comparison between our method and other state-of-the-art works on the document classification corpora. TPTSVM [18] combined transfer learning with semi-supervised learning which leveraged a large amount of unlabeled data for document classification. We can see that our method performs much better than TrAdaBoost, TPTSVM achieves the best performance when the labeled examples are not sufficient. However, ActTrAdaBoost outperforms TPTSVM when the number of instances is greater than 200. Figure 9 gives the accuracy curves of three algorithms with different numbers of iterations when the size of the target training set is 400. From the curves, we can see that our ActTrAdaBoost overcomes the negative transfer and helps to improve the transfer learning and achieves the best performance.

4. Discussion

In this work, the improved TrAdaBoost and actively transfer learning methods are proposed, and our methods achieve better performance than the existing methods. Experimental result shows that our methods can improve the transfer learning performance in PPIE task. And the reasons are analyzed as follows:

Effective RDTrAdaBoost: when the distribution from the target domain is essentially different from the source domain, and the KL distance between the source dataset and target dataset is large, TrAdaBoost tends to negative transfer. For example, in Fig. 7, the F-score of TrAdaBoost is lower than SVM when the proportion of the training data is 4%. This indicates that when the KL distance between the source dataset and target dataset is large, the source training data may contain noisy data. However, our RDTrAdaBoost can resolve the problem and we have proved it in Sect. 2. Initializing the weights by the relative distribution can increase the weights of the source domain instances with similar distribution to the target domain. As a result, the impact of domain differences on transfer learning will be reduced, therefore, RDTrAdaBoost can avoid the negative transfer. In addition, Experimental results also illustrate that RDTrAdaBoost can speed up the convergence and avoid negative transfer. Therefore the performance will be improved.

Effective ActTrAdaBoost: we propose a new actively transfer learning framework which combines active learning with the improved TrAdaBoost, namely ActTrAdaBoost. According to the defined transfer confidence measure, the instance is either labeled by the transfer classifier or directly labeled by the domain experts if needed. Therefore, the performance will be improved by leveraging experts labelling in active learning andor adding new data to enrich training data byin transfer classifier, ActTrAdaBoost can obtain better performance. The experimental results show that, when data are insufficient, the proposed ActTrAdaBoost and RDTrAdaBoost methods perform much better than the baseline SVM and original TrAdaBoost. In addition, ActTrAdaBoost outperforms TrAdaBoost when labelling cost is the same. For example, in Fig. 6 and Fig. 7, ActTrAdaBoost achieves the best results with the same labeling cost. We can see that ActTrAdaBoost is more effective than TrAdaBoost and better than RDTrAdaBoost in most cases.

5. Conclusion and Feature work

In this paper, we present an actively transfer learning framework to solve the negative transfer and lower performance problem in transfer learning. Experimental results show that the proposed ActTrAdaBoost and RDTrAdaBoost methods perform much better than the baseline SVM and original TrAdaBoost. In PPIE transfer learning task, our ActTrAdaBoost method shows better performance. Our actively transfer learning framework, not only achieves better performance with small amount of labeled data, but also provides the maximal use of the labeling process.

In recent years, deep learning has been successful in academia and industry, which is considered to be one of the most potential technologies of depth analysis of large data. Recent research has shown that deep learning can extract compact, hierarchical, abstract data representation, and has the ability to transfer and reuse across domains. Therefore, deep learning has important research value in transfer learning, and we will focus on studying new transfer learning algorithm integrating deep learning in the feature.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China under No. 61672126, 61173101, 61672127.

References

[1] Z. Yang, N. Tang, X. Zhang, H. Lin, Y. Li, and Z. Yang, Multiple kernel learning in protein-protein interaction extraction from biomedical literature, Artificial intelligence in medicine, vol.51, no.3, pp.163–173, 2011.
[2] L. Li, P. Zhang, T. Zheng, H. Zhang, Z. Jiang, and D. Huang, Integrating Semantic Information into Multiple Kernels for Protein-Protein Interaction Extraction from Biomedical Literatures, PloS one, 2014.
[3] L. Li, R. Guo, Z. Jiang, and D. Huang, "An approach to improve kernel-based Protein-Protein Interaction extraction by learning from large-scale network data," Methods, vol.83, pp.44–50, 2015.
[4] W. Fengmei, Z. Jianpei, C. Yan, and Y. Jing, "FSFP: Transfer Learning From Long Texts to the Short," Applied Mathematics & Information Sciences, vol.8, no.4, pp.2033–2040, 2014.
[5] P. Yang, W. Gao, Q. Tan, and K.-F. Wong, "A link-bridged topic model for cross-domain document classification," Information Processing & Management, vol.49, no.6, pp.1181–1193, 2013.
[6] H. Zhou, Y. Zhang, D. Huang, and L. Li, "Semi-supervised Learning with Transfer Learning," Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer Berlin Heidelberg, pp.109–119, 2013.
[7] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "A rich feature vector for protein-protein interaction extraction from multiple corpora," Association for Computaional Linguistics, Singapore: World Scientific Publishing Co Pte Ltd., pp.121–130, 2009.
[8] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," Proc. 24th international conference on Machine learning, pp.193–200, 2007.
[9] M.T. Rosenstein, Z. Marx, and L.P. Kaelbling, To transfer or not to transfer, NIPS-05 Workshop on Inductive Transfer: 10 Years Later, 2005.
[10] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," Proc. Sixteenth Annual Conference on Learning Theory, pp.825–830, 2003.
[11] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," In: Sebastiani, F. (ed.), Springer, 2003.
[12] Y. Freund, H.S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the Query By Committee algorithm," Machine Learning Journal 28, pp.133–168, 1997.
[13] M.S. Handcock and M. Morris, "Relative distribution methods," Sociological Methodology, vol.28, no.1, pp.53–97, 1998.
[14] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," Proc. twenty-first international conference on Machine learning., 2004.
[15] R. Bunescu, R. Ge, R.J. Kate, E.M. Marcotte, R.J. Mooney, A.K. Ramani, and Y.W. Wong, "Comparative experiments on learning information extractors for proteins and their interactions," Artificial intelligence in medicine, pp.139–155, 2005.
[16] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele, Mining medline: abstracts, sentences, or phrases? Proc. pacific symposium on Biocomputing, pp.326–327, 2002.
[17] K. Fundel, R. Kuffner, and R. Zimmer, "RelEx–relation extraction using dependency parse trees, Bioinformatics," vol.23, no.3, pp.365–371, 2007.
[18] H. Zhou, Y. Zhang, D. Huang, and L. Li, "Semi-supervised Learning with Transfer Learning[M]," Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer Berlin Heidelberg, pp.109–119, 2013.

Lishuang Li received her BSc degree from Dalian University of Technology in 1989, the MSc degree from Dalian University of Technology in 1992, and the PhD degree from Dalian University of Technology in 2013. She is currently a professor in School of Computer Science and Technology at Dalian University of Technology. She has published more than 60 research papers in various journals and conferences. Her research interests include text mining, natural language processing and machine translation. In recent years, she has focused on text mining for biomedical literatures and information extraction from huge text resources. Her research projects are funded by the NSFC.

Xinyu He received her BSc degree from Shenyang Normal University in 2006, and her MSc degree from Dalian University of Technology in 2012. She is an PhD candidate in the School of Computer Software and Theory at Dalian University of technology. Her research interests include text mining for biomedical literatures and information extraction from huge biomedical resources.

Jieqiong Zheng received her BSc degree from Harbin University of Science and Technology in 2014. She is an MSc candidate in School of Computer Science and Technology at Dalian University of Technology. Her research interests include information extraction from huge text resources and text mining for biomedical literatures.

Degen Huang was born in 1965. He is a professor in the Dalian University of Technology. His main research interests include natural language processing, machine learning and machine translation. He is now working at the School of Computer Science and Technology, Dalian University of Technology. He is now a senior member of CCF, and an associate editor of Int. J. Advanced Intelligence.

Fuji Ren is a Ph.D. Professor. His main research interests include Natural Language Processing, Knowledge Engineering, Sentience Computer, Machine Translation, Machine-Aided English Writing, Automatic Abstracting, Dialogue machine translation, Information Retrieval.