A Transfer Learning based Boosting Model for Emotion Analysis

Ruolan Yong, Chengyu Wang, Xiaofeng He*
Shanghai Key Laboratory of Trustworthy Computing,
School of Computer Science and Software Engineering, East China Normal University,
Shanghai, China
Email: yongruolan@126.com, chywang2013@gmail.com, xfhe@sei.ecnu.edu.cn

***Abstract*—Emotion Analysis determines the emotion of a text. Supervised Machine learning algorithms are effective for Emotion Analysis, but they need a lot of labelled data. It is a labor-intensive process and often needs instructions of experts to annotate data. In this paper, we propose a transfer learning approach for emotion analysis based on Adaboost(EATAdaBoost) by adapting the knowledge learned from labelled source data to the target domain which has none or few labelled data. We try to establish connections between source instances and target domain. Word2vec semantic similarities between source instances and common non-domain-specific emotional words which occur frequently in both domains are used as a bridge. If the similarity is bigger than a threshold, we think the source instance is useful for learning target task. In addition, we conduct extensive experiments and the results show that our algorithm is superior to baselines.**

I. Introduction

Emotion analysis aims at determining the emotions of texts automatically. It is widely applied to analyze the customers satisfaction over products [1], monitor public opinion [2], assist government agencies in making decisions [3] and so on.

An Emotion Analysis Model trained on a specific domain will have a higher generalization error if it is evaluated on a very different domain since each domain has diverse words and phrases to express emotions. For example, “There is a problem with the computer’s cooling system, it becomes very hot when used in summer.” is negative for the word “hot”. However, “hot” is positive in the context of “The story of this film is interesting and the actress is hot.” Words like “hot” which have totally opposite polarities in different domains are called “domain-specific” features.

It is time consuming if we annotate data in a new domain. Transfer learning techniques can alleviate this problem by extracting the knowledge from one or more source tasks and applying the knowledge to a target domain task. Generally speaking, there are three main kinds of transfer learning [4]: parameter-based transfer learning assumes that similar task models have analogous parameter vectors and then transfers the parameters between two domains; the intuition of instance-based transfer approach is that although instances in source domain can not be reused directly, some of them still can be reused together with a few labeled target data; feature-based approaches try to find out a “good” feature space for both domains, so that machine learning models can be trained and tested on this feature space without considering differences between domains.

TABLE I
An Example of Pivot Instances

Sentence	Emotion
现在的学生自认为打老师很厉害，实际上是缺少教养，看着真让人**生气**！（Nowdays, students think it’s cool to beat teachers. In fact, it is not educated and makes me **angry**!)	anger
从上午九点到下午两点都没有叫动他，直到我**生气**了。(He did not respond to me from 9 a.m. to 2 p.m. until I got **angry**)	anger

In this paper, we propose a transfer learning approach for e-motion analysis based on AdaBoost. In the boosting process of traditional AdaBoost, weights of wrongly predicted instances will be increased so that the next base classifier can learn the knowledge which has not been acquired by the previous base classifiers. However, in the context of transfer learning, source instances which have not been predicted correctly may be far away from the target domain. Increasing the weights constantly can mislead the base classifiers on learning too much about source noise. Under such consideration, we attempt to pick out those “useful” instances for learning target task in source domain. We assume that useful instances express emotion by common emotional words without domain specific words. They are selected by semantic similarity between instances and common emotional words which frequently occur in both domains. We call these useful instances **pivot instances**. If the pivot instances are wrongly predicted, we assign high weights to them. On the contrary, decrease the weights of instances which are not pivot instances to reduce their influence on the base classifier in next iteration.

Table I is an example of pivot instances. First sentence is an educational event-related review which will be introduced in detail in section IV. The second sentence is a description of daily life in RenCECps dataset published in [5]. Even though they are from different domains, emotion of the two sentences is only related to the common emotional word “angry” .

The main contributions on this work are as follows:

* Corresponding author.

978-1-5386-3120-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICBK.2017.31

IEEE computer society

- Propose a transfer learning algorithm based on AdaBoost with pivot instances which are selected by word2vec semantic similarities.
- Conduct comparative experiments to prove the feasibility of the algorithm.
- Carry out experiments to compare the performances of different word embedding compositions.

The rest of the paper is organized as follows: Section II introduces related work of emotion analysis, transfer learning and word representation. The algorithm we propose is presented in section III. In section IV, we will give a description of our datasets and the extensive experiments conducted. Finally, we summarize the full text and put forward the outlook to future work in section V.

II. Related Work

In this section, we will give a more detailed introduction to the related work, including emotion analysis, transfer learning and current popular feature representation methods.

A. Emotion Analysis

Emotion analysis aims at figuring out the main emotion rather than the polarity of the text. Ekam and Paul [6] summarized six basic emotions: anger, disgust, fear, happiness, sadness, and surprise. Most emotion analysis studies classified the opinion texts into the six basic emotions and the additional "none" category.

Dictionary-based methods and machine learning techniques are the classic solutions to emotion analysis. In Chinese context, Li et al. [7] classified the texts into eight classes using a semi-constructed emotion dictionary. Wu et al. [8] focused on exploring a function of emotional words annotation. Wang et al. [9] used a maximum entropy model to solve an eight classes classification problem. In addition, Wen and Wan [10] tried to mine the class sequential rules from sequential sentences.

B. Transfer Learning

Transfer learning copes with scenario in which there are none or only few labeled target data but a mount of labeled source data. TrAdaBoost [11] is the first published to solve the transfer learning problem based on AdaBoost. TrAdaBoost considers all wrongly predicted source instances are dissimilar to the target domain and decreases their weights. Shi et al. [12] pointed out that TrAdaBoost has poor performance when given improper source data, therefore Yao et al. [13] proposed MultiSurceTrAdaBoost. In addition, TransferBoost proposed by Eaton et al. [14] considers that each source domain can be a potential component of target domain distribution. Source domain which is drawn from the shared component of the target distribution could be used to augment the target training.

With the popularity of neural networks, deep learning technique is also introduced into transfer learning. Bengio [15] explained why unsupervised pre-training of representations are useful and how they can be exploited in transfer learning scenario.

C. Word Representation

"one-hot" representation is the simplest way to represent words. Each word is a N-dimension vector which only has one bit of 1 and other positions are 0. However, vocabulary size N is often very large and it is prone to cause the curse of dimensionality. In addition, this representation can not measure the semantic similarity of two word vectors.

To overcome the shortcomings of "one-hot" representation, Latent semantic analysis (LSA) [16] is proposed. It assumes that words which occur in similar context have close semantic meaning. Therefore LSA constructs a term-document matrix and the cells are the frequencies of word in the documents. Singular Value Decomposition (SVD) is used on the matrix to reduce the number of rows while preserve the similarity structure among columns.

The study of distributed word embedding makes a great process after Bengio et al. [17]. The most popular model known as word2vec [18]. Word2vec is a set of algorithms which are shallow, two-layer networks trained to reconstruct linguistic contexts of words. The semantic and syntactic patterns in word embedding of word2vec have been found can be reproduced by vector arithmetic. Patterns such as "Man is to Woman as King is to Queen" can be represented as the arithmetic operations of word vectors of "King" - "Queen" $\approx$ "Man" - "Woman".

III. Transfer learning Based on AdaBoost for Emotion Analysis

In this section, we will introduce the framework of the transfer learning algorithm, how to select pivot instances and some means of word embedding composition.

A. Transfer Learning on Emotion Analysis

An instance $x_i \in X(i = 1...m)$ is an emotional review or a text with a corresponding label y_i. $y_i \in Y(i = 1...m)$ and Y is a finite set of $|C|$ different emotional labels. The goal of emotion analysis is to learn the mapping function $f : X \to Y$ based on the training data (x_i, y_i).

However, there is often short of labeled data (x_i, y_i), all effective supervised machine learning algorithms become useless. To solve this problem, we propose a transfer learning approach for emotion analysis based on AdaBoost. Source domain S is rich in labeled training data $(x_i^s, y_i^s) \in S(i = 1...n)$. Target domain T from different distribution only has few labeled data $(x_i^t, y_i^t) \in T(i = 1...m)$. We need to learn the target mapping function f_t with auxiliary source data.

B. Framework of EATAdaBoost

As we know in transfer learning, source data is drawn from distribution different from target domain. Increasing the weights of mispredicted source instances continuously as the traditional AdaBoost do may misguide the base classifiers on learning too much noisy instances which are most likely to be far from target distribution.

In this paper, we improve AdaBoost for transfer learning. Before boosting, we try to distinguish instances useful for learning f_t from those noisy points in source domain and we

TABLE II
NOTATIONS OF TRANSFER LEARNING ON EMOTION ANALYSIS

Notation	Interpretation
x_i	An instance (a review or a text)
y_i	Emotion label of x_i
$\|C\|$	Number of categories
S	Set of source instances
T	Set of target instances
f_t	Target map function
(x_i^s, y_i^s)	Labeled source data
(x_i^t, y_i^t)	Labeled target data
p_i^s	Pivot label of (x_i^s, y_i^s)
p_i^t	Pivot label of (x_i^t, y_i^t)

call them "pivot instance". We give each instance a label p_i named *pivot label* and $p_i \in \{0,1\}$. x_i is a pivot instance only if $p_i = 1$. In next subsection, we will give a detailed introduction to how to select high-quality pivot instances. Table II is the list of notations of transfer learning for emotion analysis.

Algorithm 1 is the framework of transfer learning algorithm for emotion analysis. It is an iterative process based on AdaBoost. There are four different steps compared with AdaBoost. First, pivot labels of S and T instances are added to the input. Second, we only calculate error rate on the target labeled data to evaluate the performance of f_t^k in the $4th$ line, $I(\cdot)$ therein is an indicator function. $I(\cdot) = 1$ if the predicted label of x_i does not equal to the target label, otherwise $I(\cdot) = 0$. Third, the $5th$ line introduces a update factor β which is explained in TrAdaBoost. $exp(\beta I(\cdot)) \in (0,1]$ is multiplied to decrease the weights of noisy instances for target domain. Finally, pivot labels determine how to update the weights of instances in the 6th line. Increase weights of misclassified pivot instances by multiplying $\beta_k > 1$ and decrease the weights of non-pivot instances if they are misclassified.

The idea of our work is similar to TrAdaBoost. TrAdaBoost considers all mispredicted source instances are dissimilar to target data and decrease their weights. Our algorithm attempts to exploit semantic similarity to distinguish which instances may be beneficial for learning target classifier.

Algorithm 1 Transfer Learning based Boosting Model (EATAdaBoost).

Input:
Two labeled datasets S and T;
A base learning classifier(decision tree);
Number of iterations N;
$p_s = \{p_1^s, ..., p_n^s\}$, the pivot label set of S;
$p_t = \{p_1^t, ..., p_m^t\}$, the pivot label set of T;
$w^l = (w_1^l, ..., w_{n+m}^l)$.

Output:

$$f_t(x) = argmax_c \sum_{k=1}^{N} \beta_k I(f_t^k(x) = c);$$

1: **for** (k = 1, ... , N)
2: Normalize w_i^t. Set $w_i^k = w_i^k / (\sum_{i=1}^{n+m} w_i^k)$;
3: Generate a base classifier f_t^k trained on S and T ;
4: Calculate the error of f_t^k on T_l :

$$err_k = \sum_{i=1}^{m} \frac{w_i^k \cdot I(f_k(x_i) \neq y_i)}{\sum_{i=1}^{m} w_i^k};$$

5: Set $\beta_k = log\frac{1-err_k}{err_k} + log(|C| - 1)$,
$\beta = log\frac{1}{1+\sqrt{2lnn/N}}$
Note that, $1 - err_k$ is required to be more than $1/|C|$;
6: Update the new weight vector:

$$w_i^{k+1} = \begin{cases} w_i^k \cdot exp(\beta I(f_t^k(x_i) \neq y_i)) & p_i = 0 \\ w_i^k \cdot exp(\beta_k I(f_t^k(x_i) \neq y_i)) & p_i = 1 \end{cases};$$

Algorithm 2 Select Pivot Instances.

Input:
Labeled dataset $S = \{s_1, s_2, ..., s_n\}$;
A set of emotional words vector $E = \{e_1, e_2, ..., e_w\}$;
The threshold of semantic similarity t;
Initialize $p_s = \{p_1^s, ..., p_n^s\}$ all 0, $p_t = \{p_1^s, ..., p_m^s\}$ all 1;

Output:
$p_s = \{p_1^s, ..., p_n^s\}$, the pivot label set of S;

for (i = 1, ... , N)
  for (j = 1, ... , W)
    Calculate the semantic similarity s_{ij} of source instance s_i and emotional word e_j;
    if ($s_{ij} > t$)
      $p_i^s = 1$;
    else
      continue;

C. How to Choose Pivot Instances

We count the frequency of each emotional word and pick out the most frequent emotional words in both domains. Each emotional word is considered as an instance and represented by word2vec embedding. Algorithm 2 describes how to select the pivot instances.

We use the cosine similarity to measure the semantic similarity. It is a measure of similarity between two non zero vectors of an inner product space and the equation is $d_{cos} = \frac{x_1 \cdot x_2}{\|x_1\| \cdot \|x_2\|}$. The value of cosine similarity is between 0 and 1 and the similarity is higher when the cosine value is more closer to 1.

D. Representation of Instances

An ideal semantic space is good for representing and finding out more credible pivot instances. As we know, word2vec model only capture the semantic information from the corpus. We observe that different POS(Part of Speech) tags of a word determine the usage and the emotion polarity of the word. For example, an adjective is usually about the speaker's feeling of something or somebody. Therefore POS information is beneficial for emotion analysis. Taking this into consideration, we train another word2vec model on the words with POS tags besides the original word2vec model without POS tags.

Meanwhile this paper evaluates the performances of different means of word embedding composition, the methods are as follows:

- Average sum of word embedding: if an instance contains m words and $w_1, ..., w_m$ is the embedding of each word. Instance x_i is represented as $x_i = \sum_{i=1}^{m} w_i/m$.
- Average sum + TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) value $t_1, ..., t_m$ is taken as the weight of each word embedding, so $x_i = \sum_{i=1}^{m} t_i w_i/m$.
- SVD + word2vec[19]: create a term by document matrix and the elements of which are decided by TF-IDF values and word embedding of word2vec. Then SVD is executed on the matrix.
- Paragraph2vec [20]: paragraph2vec is the work based on word2vec. It's architecture is similar to word2vec, but it introduces a paragraph token in the input layer. This token works as a word embedding storing all the semantic information of the text. It is trained with all the pieces of texts in a slide window size.

IV. Experiments

We conduct a series of experiments on Chinese datasets to evaluate the performances of different word embedding compositions and the effectiveness of proposed algorithm. In addition, we also test the algorithm on an English multi-domain sentiment dataset.

A. Datasets

RenCECps[1] is the Chinese source data. It contains 4,004 paragraphs, 12,742 sentences and 324,571 Chinese words obtained from 500 blog articles including sina blog, sciencenet blog, baidu blog and some other websites. RenCECps is annotated at three levels: documents, sentences and features. Target data consists of about 33 million Educational-related Reviews which is clawed from the sina Weblog (WeblogER) under the education topic from January to March in 2016. The sentences in RenCECps are classified into 8 categories: *anger, anxiety, hate, joy, love, expect, sorrow, surprise* and *none*. However, there are more negative than positive reviews in WeblogER, we merge joy, love and expect into "happiness" category. The goal of the emotion analysis is to classify the WeblogER reviews into 7 categories : *happiness, anger, anxiety, hate, none, sorrow,* and *surprise* with the auxiliary dataset RenCECps.

The English corpus Multi-Domain Sentiment Dataset (version 2.0) [2] is publish in [21]. It contains four domains of product reviews: book, dvd, electronics and kitchen from Amazon. Each domain has 1000 positive and 1000 negative instances respectively. Reviews in this dataset have been preprocessed and we adopt the English pre-trained word2vec embedding [3] published by Google as the word features.

[1]http://a1-www.is.tokushima-u.ac.jp/member/ren/Ren-CECps1.0/Ren-CECps1.0.html
[2]http://www.cs.jhu.edu/ mdredze/datasets/sentiment/
[3]https://code.google.com/archive/p/word2vec/

B. Data Preprocessing of Chinese Datasets

The task is defined at sentence level. We sample 10,000 sentences from the RenCECps and annotate 3004 WeblogER instances by four annotators respectively to evaluate the transfer learning algorithm. Figure 1 is the exact number and relative

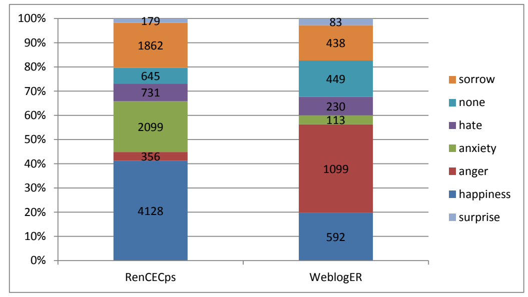

Fig. 1. Number and Ratio of Each Emotion in RenCECps and WeblogER

ratios of each category in two datasets. We can see that the most instances in the RenCECps are positive and they take up more than 40%. Conversely negative category "angry" is close to 40% in WeblogER. Data imbalance phenomenon makes it more difficult to do the classification . KL-divergence[22] of word feature space between RenCECps and WeblogER is 0.415, the distribution of one dataset more resembles the other if KL-divergence value is more close to 0.

Two Chinese word2vec models are trained on two datasets: WeblogER and 100 million entries of Baidu Encyclopedia[4] which is a knowledge sharing platform. Encyclopedia's genre and expressions are in a formal style, while WeblogER is more flexible and full of network language which increases the difficulty of word segmentation. One word2vec model is a skip-gram on the original words and the other is on the words with POS tags. The dimension of word embedding is 100, slide window size is 5.

For Chinese datasets, we calculate frequencies of each emotional word and choose the words most frequent in both datasets. 25 emotional words have been retained including "开心 (joyful)", " 幸福 (happy)", " 害怕 (fearful)" and so on. The frequencies of these words are between 10 and 140. Analogously, we select 90 English emotional words for Multi-domain Sentiment Analysis Dataset.

C. Experimental Results

To evaluate the performances of different word embedding compositions, we sample the 10,000 instances in RenCECps and 300 instances from the WeblogER according to the ratio of each category as the training data.

Table III is the performances of different word embedding compositions. All the experiments shown in table III adopt the similarity threshold value near 0.92. 1361 of 10000 source instances are selected as pivot instances. As shown in the table, distributed word embedding features are superior to manual features TF-IDF at least by 3%. Meanwhile word embedding

[4]https://baike.baidu.com/

TABLE III
THE RESULTS OF DIFFERENT WORD EMBEDDING COMPOSITIONS

Methods	Precision	Recall	MeanF1
TF-IDF	0.40	0.40	0.35
AS of Word2vec[1]	0.43	0.42	0.38
AS of TF-IDF+Word2vec	0.46	0.45	0.40
AS of SVD+word2vec[2]	0.44	0.45	0.41
AS of WPOS[3]	0.44	0.43	0.42
AS of SVD+WPOS	0.44	0.46	0.41
Paragraph2vec	0.41	0.35	0.36
Emphsis Adjective Words	**0.44**	**0.45**	**0.44**

[1] AS :Average Sum
[2] AS of SVD+word2vec: SVD is used on the average sum of word embedding matrix .
[3] WPOS: word2vec model is trained on the words with POS tagging.

trained with POS tags is more effective than the original word embedding in average. As the motivation of POS mentioned before, POS provides a kind of emotion information inherently and the results demonstrate it as expected.

In TF-IDF mode, we select 5000 features from both datasets by Chi square and take the TF-IDF as feature values.

In SVD+Word2vec, we put SVD on a 2-dimensional matrix. Each row of the matrix means a document vector composited by the average sum of the word embedding with TF-IDF weight. We keep 95% information of the singular values and the number of column decrease to 89. The result is superior to others a little. The small (100) dimension of the word embedding results in no much extra space to compress.

The paragraph2vec model initializes with word embedding generated by word2vec. Parameters of paragraph2vec are the same as word2vec. But it does not have a better performance than other methods as expected. Maybe this is because in this task, the semantic information comes the second. Lacking of training texts is also one of the reasons, paragraph2vec only takes the 13004 instances as training data.

For the better performances of the POS-related features, we conduct a simple experiment to test the impacts of POS tags on emotion analysis. Instead of assigning TF-IDF weights to each word, we adjust the weights according to POS tags. We give high weights to adjectives and low to others by the following equation:

$$w_i = \begin{cases} 2\dot{w}_i/Z & w_i\ is\ an\ adjective \\ w_i/Z & w_i\ is\ not\ an\ adjective \end{cases}$$

where Z normalizes W to be a distribution. As can be seen, Even though this method is very crude, it is very effective in emotion analysis.

We carry out the second set of experiments in Chinese and English respectively to evaluate the performances of the proposed algorithm. In Chinese data setting, all transfer algorithms adopt average sum of word embedding trained on the word2vec with POS tags as the feature space. Each algorithm is trained on 10000 RenCECps and 300 WeblogER instances and evaluated at 2704 WeblogER test instances. Results of each algorithm are shown in table IV.

TABLE IV
MEANF1SCORE OF DIFFERENT ALGORITHM FOR EMOTION ANALYSIS

Methods	Precision	Recall	MeanF1
$TrAdaboost$ (S&T)	0.40	0.43	0.39
$TransferBoost$ (S&T)	0.143	0.312	0.194
$SATdaboost$ (S&T)	**0.44**	**0.45**	**0.43**
$AdaBoost$ (S&T)	0.40	0.41	0.40
$AdaBoost_l$ (S)	0.40	0.38	0.37
$AdaBoost_h$ (T)	**0.53**	**0.55**	**0.49**

Two experiments in the grey boxes are two baselines without transfer. $AdaBoost_l(S)$ is trained only on the source data and tested on the target data, it is considered as the low baseline. $AdaBoost_h$ takes the target data as the training and test data, it works as the high baseline. Multi-class AdaBoost models uses SEMMR algorithm, base learners are decision tree with depth 4 and learning rate is 0.8.

TrAdaBoost adopts decision tree as the base classifiers and we modify it to fit the multi-class classification. The weights of instances are initialized uniformly as $1/(m+n)$, number of iteration is 500 and the procedure allow early termination. The result is not ideal since the performance depends on source domain closely.

Code of TransferBoost is shared on the webpage[5]. We adopt transfer, early termination mode, but the performance is just close to guessing. TransferBoost considers each source domain as a potential component of target domain distribution and estimates whether each source domain is drawn from target distribution by a factor α_k^i. It is maybe very effective when there are multi-source datasets, but in our experiment setting, there is only one source dataset. Increasing instances' weights in source domain by a large degree continually stresses the noisy data points. The result also shows that TransferBoost algorithm is not good at handling the imbalanced dataset problem.

The MeanF1 in table IV is near 40% on average and it is not an ideal performance. In addition to the specific features of Chinese language and diverse Internet slang, multiple emotions in a sentence also increase the difficulty of emotion analysis, such as "angry" is often accompanied by "hate". Table V is the MeanF1Score of different algorithms on English Multi-Domain Sentiment Analysis dataset. Average sum of Google English word embedding of 300-dimension is taken as the word features. Each model is trained on 1000 source and 200 target instances. The results show that our algorithm surpasses the baselines a little on average. When the kitchen reviews are the source data, there is no significance in transfer learning. It suggests that the performance of transfer learning is related to the selection of source data.

We use the Chinese datasets to test the performances of proposed algorithm with different quantities of target instances. We can see from Figure 2 that TrAdaBoost does a negative transfer learning. AdaBoost has better MeanF1Score with the increasing number of target instances. EATAdaBoost is

[5] http://www.seas.upenn.edu/ eeaton/TransferBoost/TransferBoostExp.java

TABLE V
MEANF1SCORE OF DIFFERENT ALGORITHMS ON MULTI-DOMAIN SENTIMENT ANALYSIS DATASET

	AdaBoost	EATAdaBoost	TrAdaBoost		AdaBoost	SATAdaBoost	TrAdaBoost
$book- > dvd$	0.62	**0.63**	0.61	$electronics- > book$	0.59	**0.64**	0.61
$book- > electronics$	0.62	**0.66**	0.64	$electronics- > dvd$	0.61	**0.62**	0.61
$book- > kitchen$	0.64	**0.65**	0.61	$electronics- > kitchen$	**0.66**	0.64	0.62
$dvd- > book$	0.60	**0.62**	0.59	$kitchen- > book$	**0.61**	0.60	0.59
$dvd- > electronics$	**0.69**	0.68	0.65	$kitchen- > dvd$	**0.61**	**0.61**	0.59
$dvd- > kitchen$	0.62	**0.65**	0.61	$kitchen- > electronics$	**0.71**	0.69	0.67

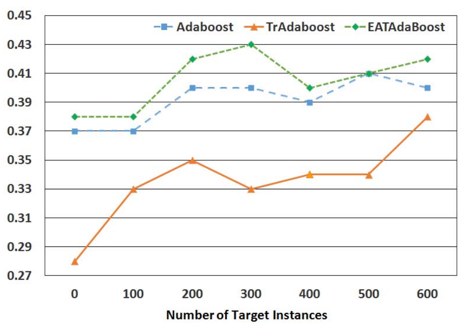

Fig. 2. MeanF1Score of Algorithms with Different Target Instances Numbers

superior to TrAdaBoost and AdaBoost on average, especially under the situation with few target instances.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose an instance-based transfer learning algorithm based on AdaBoost for emotion analysis (EATAdaboot). We conduct experiments on both Chinese and English datasets to verify the validity of the algorithm. In addition, we evaluate the performances of different word embedding compositions on Chinese corpus.

However, the results is not ideal for the imbalance multi-class classification. We will pay attention to this problem in the future.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000904.

REFERENCES

[1] D. Kang and Y. Park, “Review-based measurement of customer satisfaction in mobile service: Sentiment analysis and vikor approach,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1041–1050, 2014.
[2] I. S. Abeywardena, “Public opinion on oer and mooc: A sentiment analysis of twitter data,” in *International Conference on Open and Flexible Education*, 2014.
[3] A. Corallo and Fortunato, *Sentiment Analysis for Government: An Optimized Approach*. Springer International Publishing, 2015.
[4] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016. [Online]. Available: http://dx.doi.org/10.1186/s40537-016-0043-6
[5] C. Quan and F. Ren, “A blog emotion corpus for emotional expression analysis in chinese,” *Computer Speech and Language*, vol. 24, no. 4, pp. 726–749, 2010.
[6] P. Ekman, “An argument for basic emotions.” *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
[7] J. Li, Y. Xu, H. Xiong, and Y. Wang, “Chinese text emotion classification based on emotion dictionary,” in *IEEE Symposium on Web Society*, 2010, pp. 170 – 174.
[8] Y. Wu, K. Kita, F. Ren, K. Matsumoto, and X. Kang, “Exploring emotional words for chinese document chief emotion analysis,” in *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation, PACLIC 25, Singapore, December 16-18, 2011*, 2011, pp. 597–606. [Online]. Available: http://www.aclweb.org/anthology/Y11-1064
[9] C. Wang, C. Quan, and F. Ren, “Maximum entropy based emotion classification of chinese blog sentences,” in *International Conference on Natural Language Processing and Knowledge Engineering*, 2010, pp. 1–7.
[10] S. Wen and X. Wan, “Emotion classification in microblog texts using class sequential rules,” 2014.
[11] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Machine Learning, Proceedings of the Twenty-Fourth International Conference*, 2007, pp. 193–200.
[12] X. Shi, W. Fan, and J. Ren, “Actively transfer domain knowledge,” in *Machine Learning and Knowledge Discovery in Databases, European Conference, Ecml/pkdd 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings*, 2008, pp. 342–357.
[13] Y. Yao and G. Doretto, “Boosting for transfer learning with multiple sources,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, Ca, Usa, 13-18 June*, 2010, pp. 1855–1862.
[14] E. Eaton and M. Desjardins, “Selective transfer between learning tasks using task-based boosting,” in *AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August*, 2011, pp. 337–342.
[15] Y. Bengio, G. Guyon, V. Dror, G. Lemaire, and D. Taylor, “Deep learning of representations for unsupervised and transfer learning,” *Workshop on Unsupervised and Transfer Learning*, vol. 7, 2012.
[16] T. K. Landauer and J. Psotka, “Simulating text understanding for educational applications with latent semantic analysis: Introduction to LSA,” *Interactive Learning Environments*, vol. 8, no. 2, pp. 73–86, 2000. [Online]. Available: http://dx.doi.org/10.1076/1049-4820(200008)8:2;1-B;FT073
[17] Y. Bengio, R. Ducharme, jean, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2006.
[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *Computer Science*, 2013.
[19] R. Ju, P. Zhou, C. H. Li, and L. Liu, “An efficient method for document categorization based on word2vec and latent semantic analysis,” in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 2276–2283.
[20] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” *Computer Science*, vol. 4, pp. 1188–1196, 2014.
[21] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification,” *Acl*, vol. 31, no. 2, pp. 187–205, 2007.
[22] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 22, no. 22, pp. 79–86, 1951.