# Cross-Domain Sentiment Classification via Topic-Related TrAdaBoost

**Xingchang Huang[1], Yanghui Rao[1*], Haoran Xie[2], Tak-Lam Wong[2], Fu Lee Wang[3]**
[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2] Department of Mathematics and Information Technology, The Education University of Hong Kong,
Tai Po, New Territories, Hong Kong
[3] Caritas Institute of Higher Education, Tseung Kwan O, Hong Kong
+86-13750023601, huangxch3@mail2.sysu.edu.cn, https://github.com/xchhuang
raoyangh@mail.sysu.edu.cn, hrxie2@gmail.com, tlwong@eduhk.hk, pwang@cihe.edu.hk
*The Corresponding Author

### Abstract

Cross-domain sentiment classification aims to address the task of emotion tagging for a target domain by labeled text data from a source domain. In many online platforms, users often write their reviews on products with emotions like happiness or sadness in different domains with different words. Traditional methods usually train a classifier with sufficient labeled text data in a particular target domain. However, labeling data for each domain manually would be costly and we have to train the classifier using labeled data from other domains. Due to the distinctions between these domains, the performance of the trained classifier may be very low. In this paper, we propose a boosting-based learning framework named Topic-Related TrAdaBoost (TR-TrAdaBoost) for cross domain sentiment classification. We firstly explore the latent topic distribution of each document for the traditional TrAdaBoost, and then combine it with the unigram model. The topic distribution reveals the domain information of documents, which is valuable for cross domain sentiment classification. Thus, we extract the topic distribution of documents by topic modeling and reconstruct documents in a new feature space. Experimental results indicate that TR-TrAdaBoost represents documents well and boost the performance and robustness of the original TrAdaBoost.

## Introduction

With the rapid development of Web 2.0 services, an increasing number of users write their reviews for products they have bought after shopping online. Subjective opinions such as positive and negative are often expressed by users, which often have huge impacts on other users. Detecting the sentimental category from each user's review can help understand the perspective of users and the quality of products. Therefore, sentiment classification is widely studied to automatically predict sentimental polarity for online reviews. However, there are a great amount of reviews that are collected from different domains. It is costly to provide sufficient training text data by labeling them manually for traditional domain dependent machine learning algorithms. The sentiment classifier trained in one domain may not perform as well in other domains. The reason is that users may use domain specific words in their reviews to express sentiments in different domains, which may lead to the problem that the word distribution of documents differs in varied domains.

Due to the problem caused by domain-specific words, the TrAdaBoost (Dai et al. 2007) is developed to improve the generalization ability of classifiers. Given a large number of labeled data from the source domain ($Tr_d$), and a small amount of labeled data from the target domain ($Tr_s$), the idea of TrAdaBoost is to use boosting method to adjust the weights of training instances from $Tr_d$ and $Tr_s$. Compared to the target domain, feature distributions of $Tr_d$ are often more different than $Tr_s$. Thus, instances of $Tr_d$ should have smaller weights than $Tr_s$. However, the effectiveness of TrAdaBoost is highly dependent on $Tr_s$ if applied to cross-domain sentiment classification. For instance, the performance of TrAdaBoost decreased when $Tr_s$ mainly contain domain-independent words. This is because $Tr_s$ can not well represent the target domain with a few domain-specific words. To tackle this problem, we propose a topic-related boosting model to represent sentimental documents. The advantage of our method is that even if $Tr_s$ contains little domain-specific information, it can captures the characteristics of the target domain well by mapping word-level features to the topic-level space. Experimental results validate the effectiveness of our method on cross-domain sentiment classification.

The rest of this paper is organized as follows. We describe related work in Section 2. We present our method in Section 3. We detail the datasets, results, and discussions in Section 4. Finally, we present conclusions in Section 5.

## Related Work

Cross domain sentiment classification has drawn much attention in researchers. Zhang et al. (Zhang et al. 2014) proposed a framework to adapt two domains with the same or different emotion categories using probabilistic models. Blitzer et al. (Blitzer, McDonald, and Pereira 2006) proposed the Structural Corresponding Learning (SCL) algorithm, which adapts between the source and target domains by the selected pivot features. SCL constructs a set of tasks to model the relationship between pivot features and non-pivot features, and selects pivots according to their mutual information with the source labels (Blitzer, Dredze, and Pereira 2007). Pan et al. (Pan et al. 2010) proposed the Spectral Feature Alignment (SFA) algorithm. SFA uses some domain-independent words as a bridge to construct a bipartite graph to model the co-occurrence relationship between domain-specific words and domain-independent words. Then, SFA employs spectral clustering to cluster the domain-specific words for domain adaption. Unlike those models based on words, Li et al. (Li, Jin, and Long 2012) proposed the Topic Correlation Analysis (TCA) to group words into topics and find the correlations between these topics to bridge domains with a well-defined probabilistic topic model. TCA uses the PLSA framework and EM algorithm to extract topics and then construct the topic mapping matrix to cluster the domain-specific features.

The closely related researches are the TrAdaBoost (Dai et al. 2007) and its refinements (Cheng and Wang 2013). Given a small amount of labeled data from the target domain ($Tr_s$), the TrAdaBoost tries to decrease the weights of training instances that have different feature distributions with the target domain. However, the effectiveness of TrAdaBoost depends on the quality of $Tr_s$, and it can not deal with multiple different domains simultaneously. To jointly deal with multiple domains, Cheng et al. (Cheng and Wang 2013) proposed the weighted multi-source TrAdaBoost model. But the disturbing effect of different samples of $Tr_s$ has not been solved in an efficient way. In light of this consideration, we aim to extract the domain information for sentimental documents from both the source and the target domain by unsupervised topic modeling. Our method not only improves the performance of TrAdaBoost framework, but also adapts to multi-domain sentiment classification.

# Proposed Model

In this section, we propose our TR-TrAdaBoost for cross-domain sentiment classification. First, we give a formal definition of the research problem. Second, we describe the method of extracting topics. Third, we detail the proposed model and compare it with TrAdaBoost.

## Problem Definition

The task of cross domain sentiment classification is to build a classifier to predict the sentimental polarity of unlabeled data in a target domain based on a training set. According to the TrAdaBoost framework, the training set $Tr$ is partitioned into two labeled sets $Tr_d$ and $Tr_s$, where $Tr_d$ is collected from abundant source domains but has different feature distributions with the target domain, $Tr_s$ is a small proportion of labeled data from the target domain.

## Topic Extraction

Intuitively, users write documents with some domain-specific words in different domains. Since each document is often represented by bag-of-words or the unigram model, and the occurrence of words differs in different domains, documents from the source domain (i.e., $Tr_d$) and the target domain are varied in terms of feature distributions. Although the TrAdaBoost framework is proposed to weight instances of $Tr_d$ using labeled data from the target domain (i.e., $Tr_s$), the size of $Tr_s$ is limited and we can not ensure that $Tr_s$ represents the feature distribution of the target domain. To cope with the above problem, we extend the unigram model to a high-level latent feature space to capture the distinction between source and target domains. Besides the occurrence of words, we can observe the latent structure of topics in documents. These topics reflect the meaning of documents and contain domain-specific information, from which we can get samples of $Tr_s$ that represent the target domain well.

To extract both the domain-specific and shared topics between source and target domains, we employ the Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003). We denote the sampling topic $z_{i,j}$ as the multinomial distribution of document $i$ over topics (i.e., $\vec{\theta}_i$), and the sampling word $w_{i,j}$ as the multinomial distribution of $\vec{\varphi}_t(t = z_{i,j})$, where $z_{i,j}$ denotes the topic generated by the $i^{th}$ document and the $j^{th}$ word, and $w_{i,j}$ means the word generated by the topic of the $i^{th}$ document and the $j^{th}$ word. Further, the prior distribution of $\vec{\theta}_i$ and $\vec{\varphi}_t$ are represented as the Dirichlet distributions of $\vec{\alpha}$ and $\vec{\beta}$, respectively. Here, $\vec{\alpha}$ and $\vec{\beta}$ are hyperparameters. The generative process of LDA can be described as the following process:

1. For each topic $t \in \{1, ..., T\}$:
2. Draw $\vec{\varphi}_t \sim Dir(\vec{\beta})$
3. For each document $i \in \{1, ..., m\}$:
4. Draw $\vec{\theta}_i \sim Dir(\vec{\alpha})$
5. For all words $j$ in each document $i$:
6. Select topic index $z_{i,j} \sim Mult(\vec{\theta}_i)$
7. Select word by topic $t$: $w_{i,j} \sim Mult(\vec{\varphi}_t)|t = z_{i,j}$

In the above, $T$ means the number of topics, and $m$ means the number of documents. The above parameters of the generative model can be estimated by Gibbs sampling (Heinrich 2008), as follows:

$$\hat{\theta}_{it} = \frac{c_i^{(t)} + \alpha_t}{\sum_{t=1}^{T}(c_i^{(t)} + \alpha_t)} \tag{1}$$

$$\hat{\varphi}_{tj} = \frac{c_t^{(j)} + \beta_j}{\sum_{j=1}^{W}(c_t^{(j)} + \beta_j)} \tag{2}$$

where $c_i^{(t)}$ is the number of words in document $i$ that are assigned to topic $t$, $c_t^{(j)}$ is the number of instances of word $j$ that is assigned to topic $t$, $W$ is the total number of words. As for the hyperparameters, $\alpha_t$ and $\beta_w$ represent the document-topic association and topic-word association respectively.

## Topic-Related TrAdaBoost

After extracting topic distributions of documents, we can construct the new representation for each document. Given the document-topic matrix $\hat{\theta}_{it}$, we represent the given document $i$ as follows:

$$\phi_i = [\hat{\theta}_{i1}, \hat{\theta}_{i2}, ..., \hat{\theta}_{it}, ..., \hat{\theta}_{iT}]$$

where $\phi_i$ represents the topic distribution of the $i^{th}$ document and $\hat{\theta}_{it}$ means the topic mixture proportion on topic $t$ for document $i$. Then, we append the topic distribution $\phi_i$ to the unigram model to construct a new representation of each document. With this new representation, documents are transformed into a new feature space that is represented by both words and topics, which are more adaptive to the framework of TrAdaBoost (Dai et al. 2007) for cross-domain sentiment classification tasks.

In our TR-TrAdaBoost model, the combined training set $Tr \subset \{X \times Y\}$ is defined as follows:

$$x_i = \begin{cases} x_i^d, i = 1, ..., n \\ x_i^s, i = n+1, ..., n+m \end{cases}$$

where $x_i^d$ represents labeled data from the source domain, and $x_i^s$ denotes the small amount of labeled data from the target domain. Specially, $x_i^d$ is the auxiliary data to guild training a classifier for the target domain, and we can refine $x_i^d$ during the training process to minimize the prediction error on unlabeled data in the target domain.

---

**Algorithm 1** Topic-Related TrAdaBoost

**Input:** Labeled training data sets $Tr_d$ and $Tr_s$
Unlabeled test data set $S$
Maximum number of iterations $N$
Number of extracted topics $T$

1: Extracting the topic distribution $\phi_i$ for each document $i$ with $T$ topics.
2: Construct the new representation for each document $i$ by append the $\phi_i$ to the unigram model.
3: Initialize the weight vector $\mathbf{w}^1 = (w_1^1, ..., w_{n+m}^1)$ as random numbers.
4: For $t = 1, ..., N$ :
a. Train a basic learner using the combined data set *T*, with different weights for instances according to $w^t$. Then, get the hypothesis $h_t : X \rightarrow Y$
b. Calculate the error rate:

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t \cdot |(h_t(x_i) + c(x_i))/2|}{\sum_{i=n+1}^{n+m} w_i^t}$$

c. Let $\beta_t = \epsilon_t/(1 - \epsilon_t)$ and $\beta = 1/(1 + \sqrt{2 \ln nN})$ and assume that $\beta_t$ is less than $1/2$
d. Update the weight vector $w_i^{t+1}$ and continue the loop:

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{|(h_t(x_i)+c(x_i))/2|}, i = 1, ..., n \\ w_i^t \beta_t^{-|(h_t(x_i)+c(x_i))/2|}, i = n+1, ..., n+m \end{cases}$$

**Output:** Return the hypothesis $h_f(x)$:

$$h_f(x) = sign(\sum_{n=1}^{N} \beta_t h_t(x))$$

---

The framework of our TR-TrAdaBoost is given in Algorithm 1. For each iteration turn, we decrease the weight of a misclassified training instance (i.e., the instance of $Tr_d$ that has different feature distributions with $Tr_s$ ones) through multiplying its weight by $\beta^{|(h_t(x_i)+c(x_i))/2|}$ as $Y = \{-1, 1\}$. Therefore, the influence of the misclassified training instances will decrease in the next turn, and the instances of $Tr_d$ that fit those of $Tr_s$ better will have larger weights. The instances with large training weights will help training better classifiers for the target domain.

Compared with the original TrAdaBoost model, we can observe that our TR-TrAdaBoost introduces the parameter $T$, which represents the number of topics. The selection of $T$ should be suitable to extract the domain-specific information and domain-shared information. When we set the value of $T$ to be zero, our model is the same as the original TrAdaBoost.

## Experiments

In this section, we describe our experiments on real-world multi-domain sentiment datasets, and validate the effectiveness and improvement of our proposed model for cross-domain sentiment classification.

### Dataset

We first describe the dataset used in our experiments. The multi-domain data set is from Blitzer et al. (Blitzer, Dredze, and Pereira 2007), which contains a collection of product reviews from Amazon. Those reviews are from four product domains: books (B), dvd (D), electronics (E) and kitchen (K). Each review is assigned a sentimental polarity of positive or negative. In each domain, there are 1000 negative reviews and 1000 positive reviews, and we can construct 12 cross-domain sentiment classification tasks from them. They are D → B, E → B, K → B, K → E, D → E, B → E, B → D, K → D, E → D, B → K, D → K, E → K, where the letter before an arrow denotes the source domain and the latter one means the target domain. The processed dataset is represented as unigram and bigram features. The summary of the dataset is described in Table 1.

Table 1: Multi-Domain Sentiment Dataset

| Domain | #Reviews | #Pos | #Neg | #Features |
|---|---|---|---|---|
| books | 2,000 | 1,000 | 1,000 | 473,856 |
| dvd | 2,000 | 1,000 | 1,000 | |
| electronics | 2,000 | 1,000 | 1,000 | |
| kitchen | 2,000 | 1,000 | 1,000 | |

### Experimental Settings

In the experiments, we compare the accuracy of our model with TrAdaBoost, SVM and Topic-Related SVM. The accuracy is the percentage of correctly classified instances over all testing samples in the target domain. We denote those four algorithms as TR-TrAdaBoost, TrAdaBoost, SVM and TR-SVM respectively. SVM with linear kernel is applied in our experiments to implement SVM and TR-SVM. For TR-TrAdaBoost and TrAdaBoost, we also use the SVM with linear kernel as the basic learner. During the training of SVM learner, we always balance the training instances between positive and negative documents. We use the SVM implemented in scikit-learn (Pedregosa et al. 2012) to train those three baselines and our model. In each trained basic learner,

Table 2: Example of topic extraction in B $\rightarrow$ D task when the number of topics $T$ is 8

| Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 | Topic 6 | Topic 7 | Topic 8 |
|---|---|---|---|---|---|---|---|
| book | movie | dvd | film | book | video | book | movie |
| this_book | book | this_film | book | movie | good | film | film |
| movie | like | film | like | like | film | like | good |
| great | great | book | this_movie | really | well | read | like |
| read | film | good | love | story | movie | time | the_movie |
| like | story | like | time | this_movie | dvd | the_book | this_movie |
| the_book | people | time | movie | good | book | great | time |
| really | movie | well | characters | read | great | story | the_film |
| time | good | movie | well | love | really | better | great |
| books' | movie | great' | set' | this_book' | the_film' | season' | movie_is' |

we use a fix proportion of training instances to train the basic learner with balanced labels. For each baseline, the training instances include labeled data from the source domain $Tr_d$ and a small amount of labeled text data from the target domain $Tr_s$. For both $Tr_d$ and $Tr_s$, they also meet the constraint that the documents between positive and negative are balanced. In the following, we use them to train the classifiers for the cross-domain sentimental dataset.

The documents from the four domains are processed using the unigram and bigram model. The frequency of words in both domain that are less than 30, and the stopwords are also removed. For the proposed TR-TrAdaBoost and TR-SVM, we combine the topic representation with the original unigram and bigram model to construct the representation for documents. Further, as we use LDA (Blei, Ng, and Jordan 2003) to extract the topic distribution of documents, the hyperparameters are set to be $1/T$ ($T$ is the number of topics), where the number of topics ranges from 2 to 8 in our experiments. We use LDA for both source and target domains as a unsupervised method to extract the domain-specific topics and shared topics for the training data and construct new representation in a new feature space for documents.

For the training data, we use 2000 labeled textual data (1000 positive and 1000 negative) from the source domain as $Tr_d$ and 50 labeled textual data (25 positive and 25 negative) from the target domain as $Tr_s$ from the target domain for each task, keeping the labels being balanced. The ratio between $Tr_d$ and $Tr_s$ is set to be 0.0125 (50/2000) as Dai's work (Dai et al. 2007), which indicated that this value had great influence on the accuracy for SVM and TrAdaBoost. In the above study, when this ratio is not very large, TrAdaBoost always improves the performance of SVM. However, when the ratio is larger than 0.2, the performance of TrAdaBoost may be worse than SVM. Therefore, we set the ratio to be 0.0125 to ensure the improvement of TrAdaBoost model. For another reason, it is a more practical way to set the ratio to smaller values because we usually get insufficient labeled textual data in the target domain for supervised learning. The number of iterations is set to 50 as we find that our model can converge well after 50 iterations. The proportion of training the basic learner in boosting method is 80% of the number of training instances with balanced labels.

## Experimental Results

First, we compare the performance of the proposed TR-TrAdaBoost and TR-SVM with different numbers of topics (2 to 8), with the SVM and TrAdaBoost model in the example of B $\rightarrow$ D task. Note that we ensure the same $Tr_s$, which is sampled randomly in the target domain under different numbers of topics $T$ when we use different algorithms.

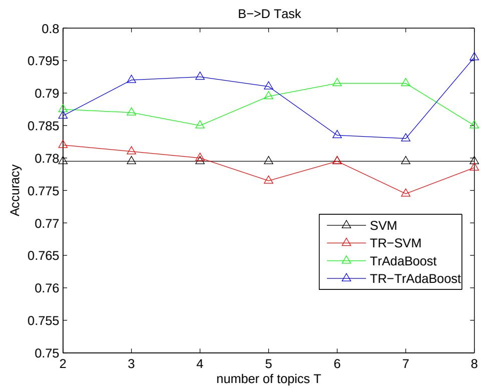


Figure 1: The accuracy curves on B $\rightarrow$ D task for four classifiers SVM, TR-SVM, TrAdaBoost and TR-TrAdaBoost

From Figure 1, we can observe the differences between these four models. SVM has no parameter relative to the number of topics $T$, so it remains the same classification accuracy. As for TR-SVM, the classification accuracy is dependent on the number of topics $T$. The Topic-Related model for SVM fluctuates without obvious improvement but with obvious decline when the number of topics $T$ equals 5 or 7. By contrast, the TrAdaBoost obviously performs better than those without boosting. Although the original TrAdaBoost model is also not relative to the number of topics $T$, its performance fluctuates because of the different sampling of training instances while training the basic learner in each iteration. The performance of our TR-TrAdaBoost model improves the classification accuracy when the number of topics $T$ is 3, 4, 5 and 8. When the number of topics $T$ is 1, 6, 7, the performance is worse than TrAdaBoost but still outperforms SVM and TR-SVM. Therefore, we can conclude that the proposed TR-TrAdaBoost makes improvements in cross domain sentiment classification tasks, with a cost of tuning the number of topics. It is interesting to observe that although our TR-TrAdaBoost introduces an extra parameter (i.e., $T$) to TrAdaBoost, it alleviate the volatility issue cause by different sampling of training samples.

Table 2 shows the result of topic extraction for B $\rightarrow$ D task when the number of topics $T$ is 8, in which the ten most related words for the specific topic are presented. For example, the "books" domain is mainly related to topic 1 as the words about books, such as "book", "this_book", "read", and "story" appear more than the words about dvds. While in topic 3, the words about dvds are more related to this topic like "dvd", "this_film", "film", "movie", and so forth. Besides, there are other words that are related to both domains and have sentimental polarity such as "like", "good", "love", "great" and so on. These domain-independent words with sentiments are important in sentiment classification tasks. Therefore, we can extract both the domain-specific information and domain-independent information, which can greatly reveals the latent features for documents and make the TrAdaBoost model more adaptive in cross domain sentiment classification. The result of B $\rightarrow$ D task with 8 topics extracted also shows the improvement of our model.

Then, we show the classification results in other tasks of these four algorithms under different numbers of topics $T$ (from 2 to 8) in Figure 2. In the B $\rightarrow$ E task, the accuracy curve is not the same with the B $\rightarrow$ D task as the best parameter $T$ for this task is 2. Similarly, the curve of TR-SVM fluctuates and its classification performance is worse that SVM. By contrast, the boosting methods with transfer learning are obviously better than others and the performance of TR-TrAdaBoost is better than TrAdaBoost when the number of topics $T$ is from 2 to 5. In B $\rightarrow$ K task, we can observe that the TR-SVM and SVM perform similarly. For these four methods, the TR-TrAdaBoost is the best one when the number of topics $T$ is 2, 3 and 7. Therefore, in this task, when the number of topics $T$ is set to be 7, our model performs best in classification accuracy. To conclude, in the B $\rightarrow$ * tasks, where the source domain is B, our model does improve the classification accuracy on the cross-domain sentiment classification tasks.

In the D $\rightarrow$ * tasks, there is some differences in the accuracy curves for these four methods. In the D $\rightarrow$ B task, the boosting methods with transfer learning (TR-TrAdaBoost and TrAdaBoost) even perform worse than others (SVM and TR-SVM) when the number of topics is 4 and 7. However, TR-TrAdaBoost and TrAdaBoost perform better in other cases, and we can observe that our model improves the accuracy obviously when the number of topics is 3 and 8. In the D $\rightarrow$ E task, it can be found that our TR-TrAdaBoost model performs better than other three methods consistently. In D $\rightarrow$ K task, the TR-TrAdaBoost does not perform well as the tasks before. However, we can observe that our model on SVM does perform better and improve a lot compared with SVM.

In the E $\rightarrow$ B task, we can also observe that the advantage of our model when the number of topics $T$ is 3, where the classification accuracy is the best among different topics. In E $\rightarrow$ D task, although the best method is TrAdaBoost when the number of topics $T$ is 7, it does not indicate that TrAdaBoost performs best in this task because its performance has no relationship with the number of topics. This is the result of the fluctuation of TrAdaBoost under different tests as the selection of training instances, while training the basic learners has great influence on the classification accuracy of TrAdaBoost. In order to present the averaged performance of TrAdaBoost, we conduct several independent tests and get the mean value as the classification accuracy, which will be shown in Table 3. The curves in the E $\rightarrow$ K task is similar to that in the D $\rightarrow$ E task because they both show the improvement of the Topic-Related model.

In the K $\rightarrow$ B task, although the TR-TrAdaBoost model does not perform well at most of the topics, it performs best when the number of topics $T$ is 2 and 7. The next task, i.e., K $\rightarrow$ D task can well demonstrate the advantage of our model. In this task, we can observe that the SVM even performs better than the TrAdaBoost model and the Topic-Related model in SVM make the classification accuracy become worse. However, our TR-TrAdaBoost model makes an obvious improvement when the number of topics $T$ is 3 and 4. This result further validates the feasibility of our method. In the last task, i.e., K $\rightarrow$ E task, we can observe that both TrAdaBoost and TR-TrAdaBoost perform similarly in average. Next, we will show detailed classification accuracy for these four methods by several independent tests.

As shown in Table 3, we present the classification accuracy for these four algorithms over these 12 cross domain sentiment classification tasks comprehensively. For SVM, we can observe that its performance is just based on the training and testing instances, regardless of the number of topics $T$. Therefore, its classification accuracy remains the same in a task under several tests. For TrAdaBoost, its performance is based on the selection of training instances while training the basic learners. Thus, its classification accuracy fluctuates under several tests. The classification accuracy of these two methods without Topic-Related model is the mean value of 10 independent tests. For those two Topics-Related model (i.e., TR-SVM and TR-TrAdaBoost), their performance are largely dependent on the selection of the number of topics. As a result, we choose the number of topics $T$ that can best improve the classification accuracy for these two methods in each task. Then, we use the mean value of 10 independent tests under the certain value of $T$ as the classification accuracy for TR-SVM and TR-TrAdaBoost. The results indicate that our TR-TrAdaBoost outperforms baselines.

## Conclusion

In this work, we propose the Topic-Related TrAdaBoost model, which makes TrAdaBoost more adaptive to the cross-domain sentiment classification tasks. The idea of our model is to extract the topic distribution of documents to capture the latent semantic structure, which contains the domain-specific and domain-shared information for documents. In our work, we consider the quality of $Tr_s$ as it is critical to cross-domain classifiers. Based on the Topic-Related model, we can improve the quality of both $Tr_s$ and $Tr_d$ by increasing the KL-divergence.

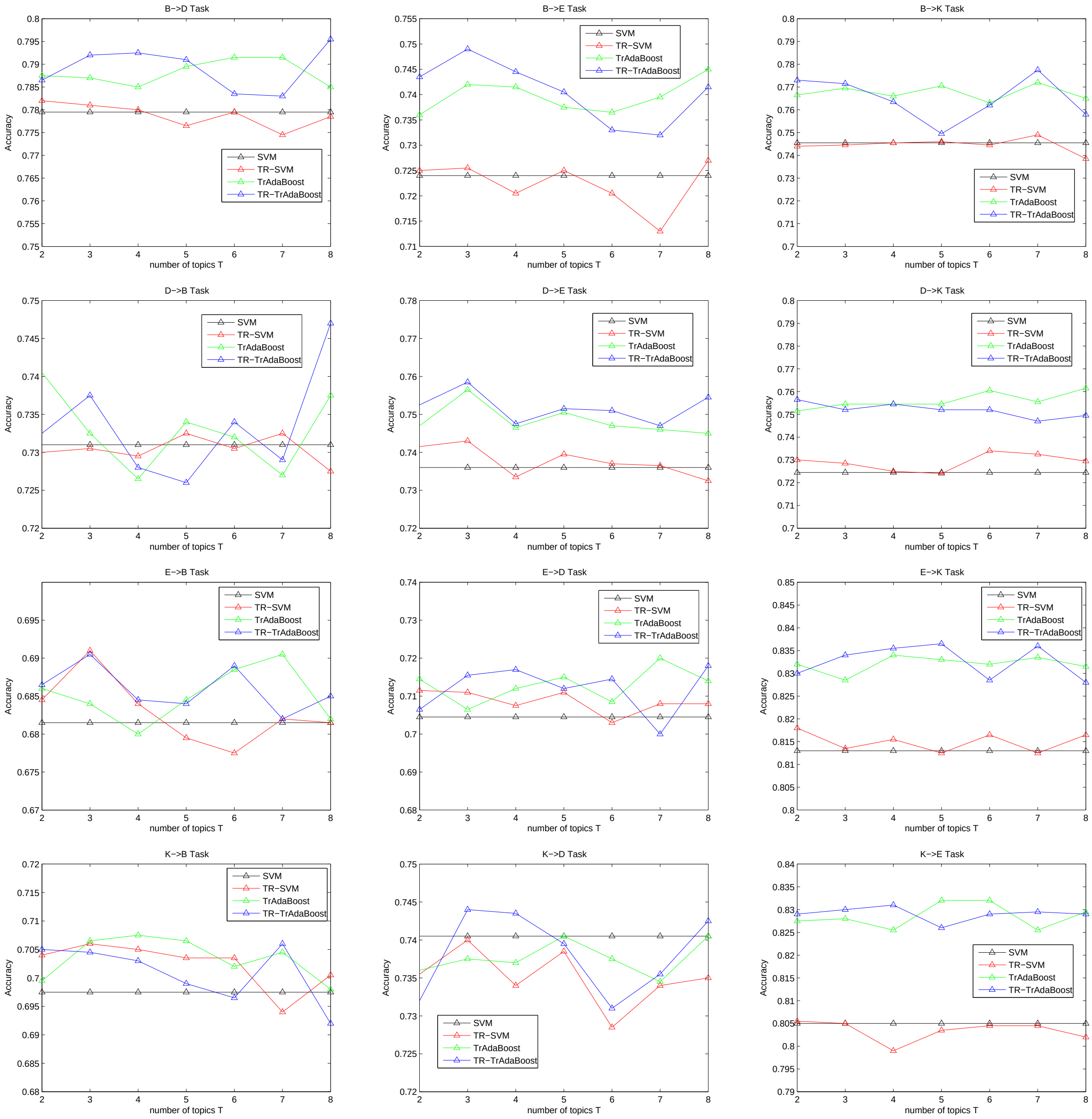


Figure 2: The accuracy curves on 12 tasks from multi-domain sentiment classification data set for four classifiers SVM, TR-SVM, TrAdaBoost and TR-TrAdaBoost with the number of topics $T$ from 2 to 8

Table 3: The classification accuracy on 12 tasks using four algorithms SVM, TR-SVM, TrAdaBoost, TR-TrAdaBoost

| Tasks | SVM | TR-SVM | TrAdaBoost | TR-TrAdaBoost |
|---|---|---|---|---|
| B $\rightarrow$ D | 0.7795 | 0.7820 | 0.7881 | **0.7955** |
| B $\rightarrow$ E | 0.7240 | 0.7270 | 0.7397 | **0.7490** |
| B $\rightarrow$ K | 0.7455 | 0.7490 | 0.7675 | **0.7775** |
| D $\rightarrow$ B | 0.7310 | 0.7325 | 0.7329 | **0.7470** |
| D $\rightarrow$ E | 0.7360 | 0.7430 | 0.7484 | **0.7585** |
| D $\rightarrow$ K | 0.7245 | 0.7340 | 0.7561 | **0.7565** |
| E $\rightarrow$ B | 0.6815 | **0.6910** | 0.6851 | 0.6905 |
| E $\rightarrow$ D | 0.7045 | 0.7115 | 0.7129 | **0.7180** |
| E $\rightarrow$ K | 0.8130 | 0.8180 | 0.8321 | **0.8365** |
| K $\rightarrow$ B | 0.6975 | **0.7060** | 0.7035 | **0.7060** |
| K $\rightarrow$ D | 0.7405 | 0.7400 | 0.7376 | **0.7440** |
| K $\rightarrow$ E | 0.8050 | 0.8055 | 0.8286 | **0.8310** |

For future work, we plan to explore the reason of performance sensitive to the number of topics $T$ for several cases. If follows that the dynamic selection of best $T$ value for different domains deserves further research.

## Acknowledgments

This research was supported by the Natural Science Foundation of China (no. 61502545), the Start-Up Research Grant (RG 37/2016-2017R) of The Education University of Hong Kong, a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS11/E06/14), and the Fundamental Research Funds for the Central Universities (no. 46000-31610009).

## References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 440–447.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 120–128.

Cheng, Y., and Wang, X. 2013. Weighted multi-source tradaboost. *Chinese Journal of Electronics* 22(3):505–510.

Dai, W.; Yang, Q.; Xue, G.-R.; and Yu, Y. 2007. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, 193–200.

Heinrich, G. 2008. Parameter estimation for text analysis. *University of Leipzig, Tech. Rep.*

Li, L.; Jin, X.; and Long, M. 2012. Topic correlation analysis for cross-domain text classification. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 998–1004.

Pan, S. J.; Ni, X.; Sun, J.-T.; Yang, Q.; and Chen, Z. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, 751–760.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; and Dubourg, V. 2012. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12(10):2825–2830.

Zhang, Y.; Zhang, N.; Si, L.; Lu, Y.; Wang, Q.; and Yuan, X. 2014. Cross-domain and cross-category emotion tagging for comments of online news. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, 627–636.