Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification

Danushka Bollegala
The University of Tokyo
7-3-1, Hongo, Tokyo,
113-8656, Japan
danushka@
iba.t.u-tokyo.ac.jp

David Weir
School of Informatics
University of Sussex
Falmer, Brighton,
BN1 9QJ, UK
d.j.weir@
sussex.ac.uk

John Carroll
School of Informatics
University of Sussex
Falmer, Brighton,
BN1 9QJ, UK
j.a.carroll@
sussex.ac.uk

Abstract

We describe a sentiment classification method that is applicable when we do not have any labeled data for a *target* domain but have some labeled data for multiple other domains, designated as the *source* domains. We automatically create a *sentiment sensitive* thesaurus using both labeled and unlabeled data from multiple source domains to find the association between words that express similar sentiments in different domains. The created thesaurus is then used to expand feature vectors to train a binary classifier. Unlike previous cross-domain sentiment classification methods, our method can efficiently learn from *multiple source* domains. Our method significantly outperforms numerous baselines and returns results that are better than or comparable to previous cross-domain sentiment classification methods on a benchmark dataset containing Amazon user reviews for different types of products.

1 Introduction

Users express opinions about products or services they consume in blog posts, shopping sites, or review sites. It is useful for both consumers as well as for producers to know what general public think about a particular product or service. Automatic document level sentiment classification (Pang et al., 2002; Turney, 2002) is the task of classifying a given review with respect to the sentiment expressed by the author of the review. For example, a sentiment classifier might classify a user review about a movie as *positive* or *negative* depending on the sentiment expressed in the review. Sentiment classification has been applied in numerous tasks such as opinion mining (Pang and Lee, 2008), opinion summarization (Lu et al., 2009), contextual advertising (Fan and Chang, 2010), and market analysis (Hu and Liu, 2004).

Supervised learning algorithms that require labeled data have been successfully used to build sentiment classifiers for a specific domain (Pang et al., 2002). However, sentiment is expressed differently in different domains, and it is costly to annotate data for each new domain in which we would like to apply a sentiment classifier. For example, in the domain of reviews about *electronics* products, the words "durable" and "light" are used to express positive sentiment, whereas "expensive" and "short battery life" often indicate negative sentiment. On the other hand, if we consider the *books* domain the words "exciting" and "thriller" express positive sentiment, whereas the words "boring" and "lengthy" usually express negative sentiment. A classifier trained on one domain might not perform well on a different domain because it would fail to learn the sentiment of the unseen words.

Work in *cross-domain sentiment classification* (Blitzer et al., 2007) focuses on the challenge of training a classifier from one or more domains (source domains) and applying the trained classifier in a different domain (target domain). A cross-domain sentiment classification system must overcome two main challenges. First, it must identify which source domain features are related to which target domain features. Second, it requires a learning framework to incorporate the information regarding the relatedness of source and target domain features. Following previous work, we define cross-domain sentiment classification as the problem of learning a binary classifier (i.e. positive or negative sentiment) given a small set of labeled data for the source domain, and unlabeled data for both source and target domains. In particular, no labeled data is provided for the target domain.

In this paper, we describe a cross-domain sentiment classification method using an automatically created sentiment sensitive thesaurus. We use labeled data from multiple source domains and unlabeled data from source and target domains to represent the distribution of features. We represent a *lexical element* (i.e. a unigram or a bigram of word lemma) in a review using a feature vector. Next, for each lexical element we measure its relatedness to other lexical elements and group related lexical elements to create a thesaurus. The thesaurus captures the relatedness among lexical elements that appear in source and target domains based on the contexts in which the lexical elements appear (their distributional context). A distinctive aspect of our approach is that, in addition to the usual co-occurrence features typically used in characterizing a word's distributional context, we make use, where possible, of the sentiment label of a document: i.e. sentiment labels form part of our context features. This is what makes the distributional thesaurus sensitive to sentiment. Unlabeled data is cheaper to collect compared to labeled data and is often available in large quantities. The use of unlabeled data enables us to accurately estimate the distribution of words in source and target domains. Our method can learn from a large amount of unlabeled data to leverage a robust cross-domain sentiment classifier.

We model the cross-domain sentiment classification problem as one of *feature expansion*, where we append additional related features to feature vectors that represent source and target domain reviews in order to reduce the mismatch of features between the two domains. Methods that use related features have been successfully used in numerous tasks such as query expansion (Fang, 2008), and document classification (Shen et al., 2009). However, feature expansion techniques have not previously been applied to the task of cross-domain sentiment classification.

In our method, we use the automatically created thesaurus to *expand* feature vectors in a binary classifier at train and test times by introducing related lexical elements from the thesaurus. We use L1 regularized logistic regression as the classification algorithm. (However, the method is agnostic to the properties of the classifier and can be used to expand feature vectors for any binary classifier). L1 regularization enables us to select a small subset of features for the classifier. Unlike previous work which attempts to learn a cross-domain classifier using a single source domain, we leverage data from multiple source domains to learn a robust classifier that generalizes across multiple domains. Our contributions can be summarized as follows.

- We describe a fully automatic method to create a thesaurus that is sensitive to the sentiment of words expressed in different domains.
- We describe a method to use the created thesaurus to expand feature vectors at train and test times in a binary classifier.

2 A Motivating Example

To explain the problem of cross-domain sentiment classification, consider the reviews shown in Table 1 for the domains *books* and *kitchen appliances*. Table 1 shows two positive and one negative review from each domain. We have emphasized in boldface the words that express the sentiment of the authors of the reviews. We see that the words **excellent**, **broad**, **high quality**, **interesting**, and **well researched** are used to express positive sentiment in the books domain, whereas the word **disappointed** indicates negative sentiment. On the other hand, in the kitchen appliances domain the words **thrilled**, **high quality**, **professional**, **energy saving**, **lean**, and **delicious** express positive sentiment, whereas the words **rust** and **disappointed** express negative sentiment. Although **high quality** would express positive sentiment in both domains, and **disappointed** negative sentiment, it is unlikely that we would encounter **well researched** in kitchen appliances reviews, or **rust** or **delicious** in book reviews. Therefore, a model that is trained only using book reviews might not have any weights learnt for **delicious** or **rust**, which would make it difficult for this model to accurately classify reviews of kitchen appliances.

	books	*kitchen appliances*
+	**Excellent** and **broad** survey of the development of civilization with all the punch of **high quality** fiction.	I was so **thrilled** when I unpack my processor. It is so **high quality** and **professional** in both looks and performance.
+	This is an **interesting** and **well researched** book.	**Energy saving** grill. My husband loves the burgers that I make from this grill. They are **lean** and **delicious**.
-	Whenever a new book by Philippa Gregory comes out, I buy it hoping to have the same experience, and lately have been sorely **disappointed**.	These knives are already showing spots of **rust** despite washing by hand and drying. Very **disappointed**.

Table 1: Positive (+) and negative (-) sentiment reviews in two different domains.

sentence	Excellent and broad survey of the development of civilization.
POS tags	Excellent/JJ and/CC broad/JJ survey/NN1 of/IO the/AT development/NN1 of/IO civilization/NN1
lexical elements (unigrams)	excellent, broad, survey, development, civilization
lexical elements (bigrams)	excellent+broad, broad+survey, survey+development, development+civilization
sentiment features (lemma)	excellent*P, broad*P, survey*P, excellent+broad*P, broad+survey*P
sentiment features (POS)	JJ*P, NN1*P, JJ+NN1*P

Table 2: Generating lexical elements and sentiment features from a positive review sentence.

3 Sentiment Sensitive Thesaurus

One solution to the feature mismatch problem outlined above is to use a thesaurus that groups different words that express the same sentiment. For example, if we know that both *excellent* and *delicious* are positive sentiment words, then we can use this knowledge to *expand* a feature vector that contains the word *delicious* using the word *excellent*, thereby reducing the mismatch between features in a test instance and a trained model. Below we describe a method to construct a sentiment sensitive thesaurus for feature expansion.

Given a labeled or an unlabeled review, we first split the review into individual sentences. We carry out part-of-speech (POS) tagging and lemmatization on each review sentence using the RASP system (Briscoe et al., 2006). Lemmatization reduces the data sparseness and has been shown to be effective in text classification tasks (Joachims, 1998). We then apply a simple word filter based on POS tags to select content words (nouns, verbs, adjectives, and adverbs). In particular, previous work has identified adjectives as good indicators of sentiment (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000). Following previous work in cross-domain sentiment classification, we model a review as a bag of words. We select unigrams and bigrams from each sentence. For the remainder of this paper, we will refer to unigrams and bigrams collectively as *lexical elements*. Previous work on sentiment classification has shown that both unigrams and bigrams are useful for training a sentiment classifier (Blitzer et al., 2007). We note that it is possible to create lexical elements both from source domain labeled reviews as well as from unlabeled reviews in source and target domains.

Next, we represent each lexical element u using a set of features as follows. First, we select other lexical elements that co-occur with u in a review sentence as features. Second, from each source domain labeled review sentence in which u occurs, we create *sentiment features* by appending the label of the review to each lexical element we generate from that review. For example, consider the sentence selected from a positive review of a book shown in Table 2. In Table 2, we use the notation "*P" to indicate positive sentiment features and "*N" to indicate negative sentiment features. The example sentence shown in Table 2 is selected from a positively labeled review, and generates positive sentiment features as shown in Table 2. In addition to word-level sentiment features, we replace words with their POS tags to create

POS-level sentiment features. POS tags generalize the word-level sentiment features, thereby reducing feature sparseness.

Let us denote the value of a feature w in the feature vector $\boldsymbol{u}$ representing a lexical element u by $f(\boldsymbol{u}, w)$. The vector $\boldsymbol{u}$ can be seen as a compact representation of the distribution of a lexical element u over the set of features that co-occur with u in the reviews. From the construction of the feature vector $\boldsymbol{u}$ described in the previous paragraph, it follows that w can be either a sentiment feature or another lexical element that co-occurs with u in some review sentence. The distributional hypothesis (Harris, 1954) states that words that have similar distributions are semantically similar. We compute $f(\boldsymbol{u}, w)$ as the pointwise mutual information between a lexical element u and a feature w as follows:

$$f(\boldsymbol{u}, w) = \log\left(\frac{\frac{c(u,w)}{N}}{\frac{\sum_{i=1}^{n} c(i,w)}{N} \times \frac{\sum_{j=1}^{m} c(u,j)}{N}}\right) \qquad (1)$$

Here, $c(u, w)$ denotes the number of review sentences in which a lexical element u and a feature w co-occur, n and m respectively denote the total number of lexical elements and the total number of features, and $N = \sum_{i=1}^{n}\sum_{j=1}^{m} c(i, j)$. Pointwise mutual information is known to be biased towards infrequent elements and features. We follow the discounting approach of Pantel & Ravichandran (2004) to overcome this bias.

Next, for two lexical elements u and v (represented by feature vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, respectively), we compute the relatedness $\tau(v, u)$ of the feature v to the feature u as follows,

$$\tau(v, u) = \frac{\sum_{w \in \{x | f(\boldsymbol{v},x)>0\}} f(\boldsymbol{u}, w)}{\sum_{w \in \{x | f(\boldsymbol{u},x)>0\}} f(\boldsymbol{u}, w)}. \qquad (2)$$

Here, we use the set notation $\{x|f(\boldsymbol{v}, x) > 0\}$ to denote the set of features that co-occur with v. Relatedness of a lexical element u to another lexical element v is the fraction of feature weights in the feature vector for the element u that also co-occur with the features in the feature vector for the element v. If there are no features that co-occur with both u and v, then the relatedness reaches its minimum value of 0. On the other hand if all features that co-occur with u also co-occur with v, then the relatedness , $\tau(v, u)$, reaches its maximum value of 1. Note that relatedness is an asymmetric measure by the definition given in Equation 2, and the relatedness $\tau(v, u)$ of an element v to another element u is not necessarily equal to $\tau(u, v)$, the relatedness of u to v.

We use the relatedness measure defined in Equation 2 to construct a *sentiment sensitive* thesaurus in which, for each lexical element u we list lexical elements v that co-occur with u (i.e. $f(\boldsymbol{u}, v) > 0$) in descending order of relatedness values $\tau(v, u)$. In the remainder of the paper, we use the term *base entry* to refer to a lexical element u for which its related lexical elements v (referred to as the *neighbors* of u) are listed in the thesaurus. Note that relatedness values computed according to Equation 2 are sensitive to sentiment labels assigned to reviews in the source domain, because co-occurrences are computed over both lexical and sentiment elements extracted from reviews. In other words, the relatedness of an element u to another element v depends upon the sentiment labels assigned to the reviews that generate u and v. This is an important fact that differentiates our sentiment-sensitive thesaurus from other distributional thesauri which do not consider sentiment information.

Moreover, we only need to retain lexical elements in the sentiment sensitive thesaurus because when predicting the sentiment label for target reviews (at test time) we cannot generate sentiment elements from those (unlabeled) reviews, therefore we are not required to find expansion candidates for sentiment elements. However, we emphasize the fact that the relatedness values between the lexical elements listed in the sentiment-sensitive thesaurus are computed using co-occurrences with both lexical and sentiment features, and therefore the expansion candidates selected for the lexical elements in the target domain reviews are sensitive to sentiment labels assigned to reviews in the source domain. Using a sparse matrix format and approximate similarity matching techniques (Sarawagi and Kirpal, 2004), we can efficiently create a thesaurus from a large set of reviews.

4 Feature Expansion

Our*feature expansion* phase augments a feature vector with additional related features selected from the sentiment-sensitive thesaurus created in Section 3 to overcome the feature mismatch problem. First, following the bag-of-words model, we model a review d using the set $\{w_1, \ldots, w_N\}$, where the elements w_i are either unigrams or bigrams that appear in the review d. We then represent a review d by a real-valued term-frequency vector $\boldsymbol{d} \in \mathbb{R}^N$, where the value of the j-th element d_j is set to the total number of occurrences of the unigram or bigram w_j in the review d. To find the suitable candidates to expand a vector $\boldsymbol{d}$ for the review d, we define a ranking score $\mathrm{score}(u_i, \boldsymbol{d})$ for each base entry in the thesaurus as follows:

$$\mathrm{score}(u_i, \boldsymbol{d}) = \frac{\sum_{j=1}^{N} d_j \tau(w_j, u_i)}{\sum_{l=1}^{N} d_l} \tag{3}$$

According to this definition, given a review d, a base entry u_i will have a high ranking score if there are many words w_j in the review d that are also listed as neighbors for the base entry u_i in the sentiment-sensitive thesaurus. Moreover, we weight the relatedness scores for each word w_j by its normalized term-frequency to emphasize the salient unigrams and bigrams in a review. Recall that relatedness is defined as an asymmetric measure in Equation 2, and we use $\tau(w_j, u_i)$ in the computation of $\mathrm{score}(u_i, \boldsymbol{d})$ in Equation 3. This is particularly important because we would like to score base entries u_i considering *all* the unigrams and bigrams that appear in a review d, instead of considering each unigram or bigram individually.

To expand a vector, $\boldsymbol{d}$, for a review d, we first rank the base entries, u_i using the ranking score in Equation 3 and select the top k ranked base entries. Let us denote the r-th ranked ($1 \leq r \leq k$) base entry for a review d by v_d^r. We then extend the original set of unigrams and bigrams $\{w_1, \ldots, w_N\}$ by the base entries $v_d^1, \ldots, v_d^k$ to create a new vector $\boldsymbol{d'} \in \mathbb{R}^{(N+k)}$ with dimensions corresponding to $w_1, \ldots, w_N, v_d^1, \ldots, v_d^k$ for a review d. The values of the extended vector $\boldsymbol{d'}$ are set as follows. The values of the first N dimensions that correspond to unigrams and bigrams w_i that occur in the review d are set to d_i, their frequency in d. The subsequent k dimensions that correspond to the top ranked based entries for the review d are weighted according to their ranking score. Specifically, we set the value of the r-th ranked base entry v_d^r to $1/r$. Alternatively, one could use the ranking score, $\mathrm{score}(v_d^r, d)$, itself as the value of the appended base entries. However, both relatedness scores as well as normalized term-frequencies can be small in practice, which leads to very small absolute ranking scores. By using the inverse rank, we only take into account the relative ranking of base entries and ignore their absolute scores.

Note that the score of a base entry depends on a review d. Therefore, we select different base entries as additional features for expanding different reviews. Furthermore, we do *not* expand each w_i individually when expanding a vector $\boldsymbol{d}$ for a review. Instead, we consider *all* unigrams and bigrams in d when selecting the base entries for expansion. One can think of the feature expansion process as a lower dimensional latent mapping of features onto the space spanned by the base entries in the sentiment-sensitive thesaurus. The asymmetric property of the relatedness (Equation 2) implicitly prefers common words that co-occur with numerous other words as expansion candidates. Such words act as domain independent pivots and enable us to transfer the information regarding sentiment from one domain to another.

Using the extended vectors $\boldsymbol{d'}$ to represent reviews, we train a binary classifier from the source domain labeled reviews to predict positive and negative sentiment in reviews. We differentiate the appended base entries v_d^r from w_i that existed in the original vector $\boldsymbol{d}$ (prior to expansion) by assigning different feature identifiers to the appended base entries. For example, a unigram *excellent* in a feature vector is differentiated from the base entry *excellent* by assigning the feature id, "BASE=*excellent*" to the latter. This enables us to learn different weights for base entries depending on whether they are useful for expanding a feature vector. We use L1 regularized logistic regression as the classification algorithm (Ng, 2004), which produces a sparse model in which most irrelevant features are assigned a zero weight. This enables us to select useful features for classification in a systematic way without having to preselect features using heuristic approaches. The regularization parameter is set to its default value of 1 for all the experiments described in this paper.

5 Experiments

5.1 Dataset

To evaluate our method we use the cross-domain sentiment classification dataset prepared by Blitzer et al. (2007). This dataset consists of Amazon product reviews for four different product types: books (**B**), DVDs (**D**), electronics (**E**) and kitchen appliances (**K**). There are 1000 positive and 1000 negative labeled reviews for each domain. Moreover, the dataset contains some unlabeled reviews (on average $17,547$) for each domain. This benchmark dataset has been used in much previous work on cross-domain sentiment classification and by evaluating on it we can directly compare our method against existing approaches.

Following previous work, we randomly select 800 positive and 800 negative labeled reviews from each domain as training instances (i.e. $1600 \times 4 = 6400$); the remainder is used for testing (i.e. $400 \times 4 = 1600$). In our experiments, we select each domain in turn as the target domain, with one or more other domains as sources. Note that when we combine more than one source domain we limit the total number of source domain labeled reviews to 1600, balanced between the domains. For example, if we combine two source domains, then we select 400 positive and 400 negative labeled reviews from each domain giving $(400 + 400) \times 2 = 1600$. This enables us to perform a fair evaluation when combining multiple source domains. The evaluation metric is classification accuracy on a target domain, computed as the percentage of correctly classified target domain reviews out of the total number of reviews in the target domain.

5.2 Effect of Feature Expansion

To study the effect of feature expansion at train time compared to test time, we used Amazon reviews for two further domains, *music* and *video*, which were also collected by Blitzer et al. (2007) but are not part of the benchmark dataset. Each validation domain has 1000 positive and 1000 negative labeled reviews, and 15000 unlabeled reviews. Using the validation domains as targets, we vary the number of top k ranked base entries (Equation 3) used for feature expansion during training (Train_k) and testing (Test_k), and measure the average classification accuracy. Figure 1 illustrates the results using a *heat map*, where dark colors indicate low accuracy values and light colors indicate high accuracy values. We see that expanding features only at test time (the left-most column) does not work well because we have not learned proper weights for the additional features. Similarly, expanding features only at train time (the bottom-most row) also does not perform well because the expanded features are not used during testing. The maximum classification accuracy is obtained when $\mathrm{Test}_k = 400$ and $\mathrm{Train}_k = 800$, and we use these values for the remainder of the experiments described in the paper.

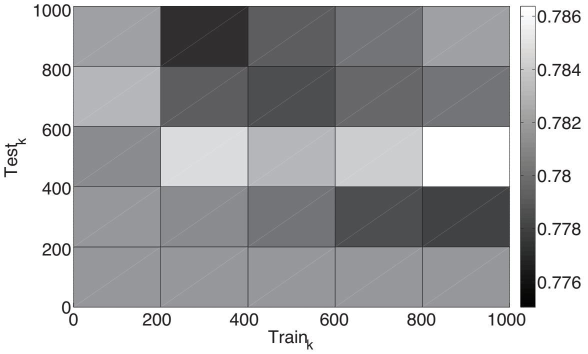

Figure 1: Feature expansion at train vs. test times.

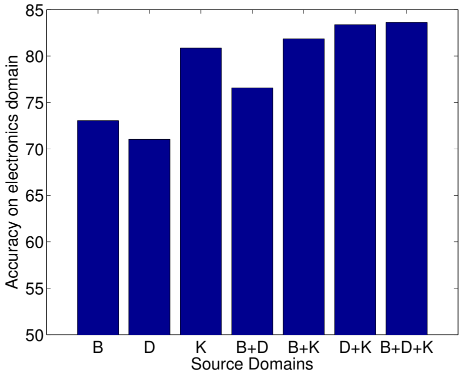

Figure 2: Effect of using multiple source domains.

5.3 Combining Multiple Sources

Figure 2 shows the effect of combining multiple source domains to build a sentiment classifier for the electronics domain. We see that the kitchen domain is the single best source domain when adapting to the electronics target domain. This behavior

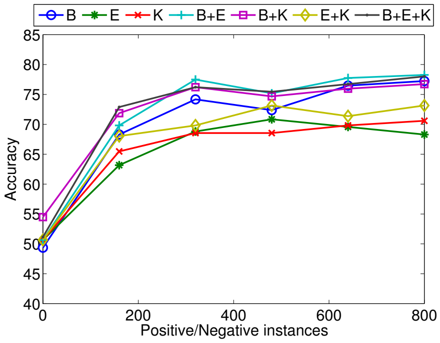

Figure 3: Effect of source domain labeled data.

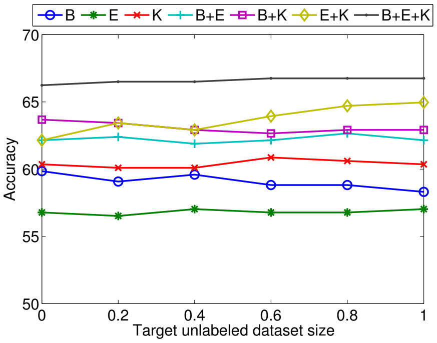

Figure 5: Effect of target domain unlabeled data.

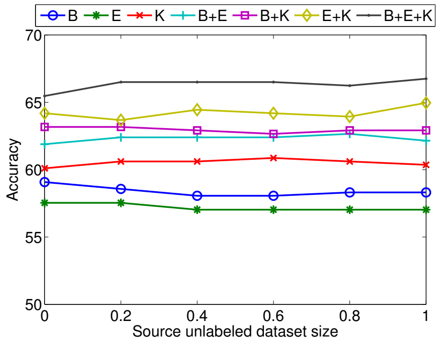

Figure 4: Effect of source domain unlabeled data.

is explained by the fact that in general kitchen appliances and electronic items have similar aspects. But a more interesting observation is that the accuracy that we obtain when we use two source domains is always greater than the accuracy if we use those domains individually. The highest accuracy is achieved when we use all three source domains. Although not shown here for space limitations, we observed similar trends with other domains in the benchmark dataset.

To investigate the impact of the quantity of source domain labeled data on our method, we vary the amount of data from zero to 800 reviews, with equal amounts of positive and negative labeled data. Figure 3 shows the accuracy with the DVD domain as the target. Note that source domain labeled data is used both to create the sentiment sensitive thesaurus as well as to train the sentiment classifier. When there are multiple source domains we limit and balance the number of labeled instances as outlined in Section 5.1. The amount of unlabeled data is held constant, so that any change in classification accuracy is directly attributable to the source domain labeled instances. Because this is a binary classification task (i.e. positive vs. negative sentiment), a random classifier that does not utilize any labeled data would report a 50% classification accuracy. From Figure 3, we see that when we increase the amount of source domain labeled data the accuracy increases quickly. In fact, by selecting only 400 (i.e. 50% of the total 800) labeled instances per class, we achieve the maximum performance in most of the cases.

To study the effect of source and target domain unlabeled data on the performance of our method, we create sentiment sensitive thesauri using different proportions of unlabeled data. The amount of labeled data is held constant and is balanced across multiple domains as outlined in Section 5.1, so any changes in classification accuracy can be directly attributed to the contribution of unlabeled data. Figure 4 shows classification accuracy on the DVD target domain when we vary the proportion of source domain unlabeled data (target domain's unlabeled data is fixed).

Likewise, Figure 5 shows the classification accuracy on the DVD target domain when we vary the proportion of the target domain's unlabeled data (source domains' unlabeled data is fixed). From Figures 4 and 5, we see that irrespective of the amount being used, there is a clear performance gain when we use unlabeled data from multiple source domains compared to using a single source domain. However, we could not observe a clear gain in performance when we increase the amount of the unlabeled data used to create the sentiment sensitive thesaurus.

Method	K	D	E	B
No Thesaurus	72.61	68.97	70.53	62.72
SCL	80.83	74.56	78.43	72.76
SCL-MI	82.06	76.30	78.93	74.56
SFA	81.48	76.31	75.30	**77.73**
LSA	79.00	73.50	77.66	70.83
FALSA	80.83	76.33	77.33	73.33
NSS	77.50	73.50	75.50	71.46
Proposed	**85.18**	**78.77**	**83.63**	76.32
Within-Domain	*87.70*	*82.40*	*84.40*	*80.40*

Table 3: Cross-domain sentiment classification accuracy.

5.4 Cross-Domain Sentiment Classification

Table 3 compares our method against a number of baselines and previous cross-domain sentiment classification techniques using the benchmark dataset. For all previous techniques we give the results reported in the original papers. The **No Thesaurus** baseline simulates the effect of not performing any feature expansion. We simply train a binary classifier using unigrams and bigrams as features from the labeled reviews in the source domains and apply the trained classifier on the target domain. This can be considered to be a lower bound that does not perform domain adaptation. **SCL** is the structural correspondence learning technique of Blitzer et al. (2006). In **SCL-MI**, features are selected using the mutual information between a feature (unigram or bigram) and a domain label. After selecting salient features, the SCL algorithm is used to train a binary classifier. **SFA** is the spectral feature alignment technique of Pan et al. (2010). Both the **LSA** and **FALSA** techniques are based on latent semantic analysis (Pan et al., 2010). For the **Within-Domain** baseline, we train a binary classifier using the labeled data from the target domain. This upper baseline represents the classification accuracy we could hope to obtain if we were to have labeled data for the target domain. Note that this is not a cross-domain classification setting. To evaluate the benefit of using sentiment features on our method, we give a **NSS** (non-sentiment sensitive) baseline in which we create a thesaurus without using any sentiment features. **Proposed** is our method.

From Table 3, we see that our **proposed** method returns the best cross-domain sentiment classification accuracy (shown in boldface) for the three domains kitchen appliances, DVDs, and electronics. For the books domain, the best results are returned by **SFA**. The books domain has the lowest number of unlabeled reviews (around 5000) in the dataset. Because our method relies upon the availability of unlabeled data for the construction of a sentiment sensitive thesaurus, we believe that this accounts for our lack of performance on the books domain. However, given that it is much cheaper to obtain unlabeled than labeled data for a target domain, there is strong potential for improving the performance of our method in this domain. The analysis of variance (ANOVA) and Tukey's honestly significant differences (HSD) tests on the classification accuracies for the four domains show that our method is statistically significantly better than both the **No Thesaurus** and **NSS** baselines, at confidence level 0.05. We therefore conclude that using the sentiment sensitive thesaurus for feature expansion is useful for cross-domain sentiment classification. The results returned by our method are comparable to state-of-the-art techniques such as **SCL-MI** and **SFA**. In particular, the differences between those techniques and our method are not statistically significant.

6 Related Work

Compared to single-domain sentiment classification, which has been studied extensively in previous work (Pang and Lee, 2008; Turney, 2002), cross-domain sentiment classification has only recently received attention in response to advances in the area of domain adaptation. Aue and Gammon (2005) report a number of empirical tests into domain adaptation of sentiment classifiers using an ensemble of classifiers. However, most of these tests were unable to outperform a simple baseline classifier that is trained using all labeled data for all domains.

Blitzer et al. (2007) apply the structural correspondence learning (SCL) algorithm to train a cross-domain sentiment classifier. They first chooses a set of *pivot features* using pointwise mutual information between a feature and a domain label. Next, linear predictors are learnt to predict the occurrences of those pivots. Finally, they use singular value decomposition (SVD) to construct a lower-dimensional feature space in which a binary classifier is trained. The selection of pivots is vital to the performance of SCL and heuristically selected pivot features might not guarantee the best performance on target domains. In contrast, our method uses all features when creating the thesaurus and selects a subset of features during training using L1 regularization. Moreover, we do not require SVD, which has cubic time complexity so can be computationally expensive for large datasets.

Pan et al. (2010) use structural feature alignment (SFA) to find an alignment between domain specific and domain independent features. The mutual information of a feature with domain labels is used to classify domain specific and domain independent features. Next, spectral clustering is performed on a bipartite graph that represents the relationship between the two sets of features. Finally, the top eigenvectors are selected to construct a lower-dimensional projection. However, not all words can be cleanly classified into domain specific or domain independent, and this process is conducted prior to training a classifier. In contrast, our method lets a particular lexical entry to be listed as a neighour for multiple base entries. Moreover, we expand each feature vector individually and do not require any clustering. Furthermore, unlike SCL and SFA, which consider a single source domain, our method can efficiently adapt from multiple source domains.

7 Conclusions

We have described and evaluated a method to construct a sentiment-sensitive thesaurus to bridge the gap between source and target domains in cross-domain sentiment classification using multiple source domains. Experimental results using a benchmark dataset for cross-domain sentiment classification show that our proposed method can improve classification accuracy in a sentiment classifier. In future, we intend to apply the proposed method to other domain adaptation tasks.

Acknowledgement

This research was conducted while the first author was a visiting research fellow at Sussex university under the postdoctoral fellowship of the Japan Society for the Promotion of Science (JSPS).

References

Anthony Aue and Michael Gamon. 2005. Customizing sentiment classifiers to new domains: a case study. Technical report, Microsoft Research.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP 2006*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL 2007*, pages 440–447.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the rasp system. In *COLING/ACL 2006 Interactive Presentation Sessions*.

Teng-Kai Fan and Chia-Hui Chang. 2010. Sentiment-oriented contextual advertising. *Knowledge and Information Systems*, 23(3):321–344.

Hui Fang. 2008. A re-examination of query expansion using lexical resources. In *ACL 2008*, pages 139–147.

Z. Harris. 1954. Distributional structure. *Word*, 10:146–162.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *ACL 1997*, pages 174–181.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD 2004*, pages 168–177.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *ECML 1998*, pages 137–142.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *WWW 2009*, pages 131–140.

Andrew Y. Ng. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML 2004*.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *WWW 2010*.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP 2002*, pages 79–86.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *NAACL-HLT'04*, pages 321 – 328.

Sunita Sarawagi and Alok Kirpal. 2004. Efficient set joins on similarity predicates. In *SIGMOD '04*, pages 743–754.

Dou Shen, Jianmin Wu, Bin Cao, Jian-Tao Sun, Qiang Yang, Zheng Chen, and Ying Li. 2009. Exploiting term relationship to boost text classification. In *CIKM'09*, pages 1637 – 1640.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL 2002*, pages 417–424.

Janyce M. Wiebe. 2000. Learning subjective adjective from corpora. In *AAAI 2000*, pages 735–740.