Cross-domain sentiment classification using a sentiment sensitive thesaurus

Article (Accepted Version)

Cross-Domain Sentiment Classification using a Sentiment Sensitive Thesaurus

Danushka Bollegala, *Member, IEEE,* David Weir and John Carroll

Abstract—Automatic classification of sentiment is important for numerous applications such as opinion mining, opinion summarization, contextual advertising, and market analysis. Typically, sentiment classification has been modeled as the problem of training a binary classifier using reviews annotated for positive or negative sentiment. However, sentiment is expressed differently in different domains, and annotating corpora for every possible domain of interest is costly. Applying a sentiment classifier trained using labeled data for a particular domain to classify sentiment of user reviews on a different domain often results in poor performance because words that occur in the train (source) domain might not appear in the test (target) domain. We propose a method to overcome this problem in cross-domain sentiment classification. First, we create a sentiment sensitive distributional thesaurus using labeled data for the source domains and unlabeled data for both source and target domains. Sentiment sensitivity is achieved in the thesaurus by incorporating document level sentiment labels in the context vectors used as the basis for measuring the distributional similarity between words. Next, we use the created thesaurus to *expand* feature vectors during train and test times in a binary classifier. The proposed method significantly outperforms numerous baselines and returns results that are comparable with previously proposed cross-domain sentiment classification methods on a benchmark dataset containing Amazon user reviews for different types of products. We conduct an extensive empirical analysis of the proposed method on single and multi-source domain adaptation, unsupervised and supervised domain adaptation, and numerous similarity measures for creating the sentiment sensitive thesaurus. Moreover, our comparisons against the SentiWordNet, a lexical resource for word polarity, show that the created sentiment-sensitive thesaurus accurately captures words that express similar sentiments.

Index Terms—Cross-Domain Sentiment Classification, Domain Adaptation, Thesauri Creation

1 Introduction

Users express their opinions about products or services they consume in blog posts, shopping sites, or review sites. Reviews on a wide variety of commodities are available on the Web such as, books (amazon.com), hotels (tripadvisor.com), movies (imdb.com), automobiles (caranddriver.com), and restaurants (yelp.com). It is useful for both the consumers as well as for the producers to know what general public think about a particular product or service. Automatic document level sentiment classification [1], [2] 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 [3], opinion summarization [4], contextual advertising [5], and market analysis [6]. For example, in an opinion summarization system it is useful to first classify all reviews into positive or negative sentiments and then create a summary for each sentiment type for a particular product. A contextual advert placement system might decide to display an advert for a particular product if a positive sentiment is expressed in a blog post.

• *D. Bollegala is with University of Tokyo, danushka@iba.t.u-tokyo.ac.jp*
D. Weir and J. Carroll are with University of Sussex, {j.a.carroll,d.j.weir}@sussex.ac.uk

Supervised learning algorithms that require labeled data have been successfully used to build sentiment classifiers for a given domain [1]. 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 *electronics* domain 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 fails to learn the sentiment of the unseen words.

The *cross-domain sentiment classification* problem [7], [8] focuses on the challenge of training a classifier from one or more domains (source domains) and applying the trained classifier on a different domain (target domain). A cross-domain sentiment classification system must overcome two main challenges. First, we must identify which source domain features are related to which target domain features. Second, we require a learning framework to incorporate the information regarding the relatedness of source and target domain features. In this paper, we propose a cross-domain sentiment classification method that overcomes both those challenges.

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 mis-match of features between the two domains. Methods that use related features have been successfully used in numerous tasks such as query expansion [9] in information retrieval [10], and document classification [11]. For example, in query expansion, a user query containing the word *car* might be expanded to *car OR automobile*, thereby retrieving documents that contain either the term *car* or the term *automobile*. However, to the best of our knowledge, feature expansion techniques have not previously been applied to the task of cross-domain sentiment classification.

We create a sentiment sensitive thesaurus that aligns different words that express the same sentiment in different domains. We use labeled data from multiple source domains and unlabeled data from source and target domains to represent the distribution of features. We use *lexical elements* (unigrams and bigrams of word lemma) and *sentiment elements* (rating information) to represent a user review. Next, for each lexical element we measure its relatedness to other lexical elements and group related lexical elements to create a sentiment sensitive 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 (its 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 sentiment sensitive. 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. The proposed method can learn from a large amount of unlabeled data to leverage a robust cross-domain sentiment classifier.

In our proposed 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 proposed method is agnostic to the properties of the classifier and can be used to expand feature vectors for any binary classifier. As shown later in the experiments, L1 regularization enables us to select a small subset of features for the classifier.

Our contributions in this work can be summarized as follows.

- We propose a fully automatic method to create a thesaurus that is sensitive to the sentiment of words expressed in different domains. We utilize both labeled and unlabeled data available for the source domains and unlabeled data from the target domain.
- We propose a method to use the created thesaurus to expand feature vectors at train and test times in a binary classifier.
- We compare the sentiment classification accuracy of our proposed method against numerous baselines and previously proposed cross-domain sentiment classification methods for both single source and multi-source adaptation settings.
- We conduct a series of experiments to evaluate the potential applicability of the proposed method in real-world domain adaptation settings. The performance of the proposed method directly depends on the sentiment sensitive thesaurus we use for feature expansion. In Section 6.3, we create multiple thesauri using different relatedness measures and study the level of performance achieved by the proposed method. In real-world settings we usually have numerous domain at our disposal that can be used as sources to adapt to a novel target domain. Therefore, it is important to study how the performance of the proposed method vary when we have multiple source domains. We study this effect experimentally in Section 6.4. The amount of training data required by a domain adaptation method to achieve an acceptable level of performance on a target domain is an important factor. In Section 6.5, we experimentally study the effect on source/target labeled/unlabeled dataset sizes on the proposed method.
- We study the ability of our method to accurately predict the polarity of words using SentiWordNet, a lexical resource in which each WordNet synset is associated with a polarity score.

2 Problem Setting

We define a domain D as a class of entities in the world or a semantic concept. For example, different types of products such as books, DVDs, or automobiles are considered as different domains. Given a review written by a user on a product that belongs to a particular domain, the objective is to predict the sentiment expressed by the author in the review about the product. We limit ourselves to binary sentiment classification of entire reviews.

We denote a source domain by D_{src} and a target domain by D_{tar}. The set of labeled instances from the source domain, $L(D_{src})$, contains pairs $(\boldsymbol{t}, c)$ where a review, $\boldsymbol{t}$, is assigned a sentiment label, c. Here, $c \in \{1, -1\}$, and the sentiment labels $+1$ and -1 respectively denote positive and negative sentiments. In addition to positive and negative sentiment reviews, there can also be *neutral* and *mixed* reviews in practical applications. If a review discusses both positive and negative aspects of a particular product, then such a review is considered as a mixed sentiment review. On the other hand, if a review does not contain neither positive nor negative sentiment regarding a particular product then it is considered as neutral. Although this paper only focuses on

positive and negative sentiment reviews, it is not hard to extend the proposed method to address multi-category sentiment classification problems.

In addition to the labeled data from the source domain, we assume the availability of unlabeled data from both source and target domains. We denote the set of unlabeled data in the source domain by $U(D_{src})$, and the set of unlabeled data in the target domain by $U(D_{tar})$. We define cross-domain sentiment classification as the task of learning a binary classifier, F using $L(D_{src})$, $U(D_{src})$, and $U(D_{tar})$ to predict the sentiment label of a review $\boldsymbol{t}$ in the target domain. Unlike previous work which attempts to learn a cross-domain classifier using a single source domain, we use data from multiple source domains to learn a robust classifier that generalizes across multiple domains.

3 A Motivating Example

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

One solution to this feature mismatch problem 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. There are two important questions that must be addressed in this approach: *how to automatically construct a thesaurus that is sensitive to the sentiments expressed by words?*, and *how to use the thesaurus to expand feature vectors during training and classification?*. The first question is discussed in Section 4, where we propose a distributional approach to construct a sentiment sensitive thesaurus using both labeled and unlabeled data from multiple domains. The second question is addressed in Section 5, where we propose a ranking score to select the candidates from the thesaurus to expand a given feature vector.

4 Sentiment Sensitive Thesaurus

As we saw in our example in Section 3, a fundamental problem when applying a sentiment classifier trained on a particular domain to classify reviews on a different domain is that words (hence features) that appear in the reviews in the target domain do not always appear in the trained model. To overcome this feature mismatch problem, we construct a sentiment sensitive thesaurus that captures the relatedness of words as used in different domains. Next, we describe the procedure to construct our sentiment sensitive thesaurus.

Given a labeled or an unlabeled review, we first split the review into individual sentences and conduct part-of-speech (POS) tagging and lemmatization using the RASP system [12]. Lemmatization is the process of normalizing the inflected forms of a word to its lemma. For example, both singular and plural versions of a noun are lemmatized to the same base form. Lemmatization reduces the feature sparseness and has shown to be effective in text classification tasks [13].

We then apply a simple word filter based on POS tags to filter out function words, retaining only nouns, verbs, adjectives, and adverbs. In particular, adjectives have been identified as good indicators of sentiment in previous work [14], [15]. Following the previous work in cross-domain sentiment classification, we model a review as a bag of words. We then select unigrams and bigrams from each sentence. For the remainder of this paper, we will refer both unigrams and bigrams collectively as *lexical* elements. In previous work on sentiment classification it has been shown that the use of both unigrams and bigrams are useful to train a sentiment classifier [7]. We note that it is possible to create lexical elements from both source domain labeled reviews ($L(D_{src})$) as well as unlabeled reviews from source and target domains ($U(D_{src})$ and $U(D_{tar})$).

Next, from each source domain labeled review we create *sentiment* elements 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 on a book shown in Table 2. In Table 2, we use the notation "*P" to indicate positive sentiment elements and "*N" to indicate negative sentiment elements. The example sentence shown in Table 2 is selected from a positively labeled review, and generates positive sentiment elements as show in Table 2. Sentiment elements, extracted only using labeled reviews in the source domain, encode the sentiment information for lexical elements extracted from source and target domains.

We represent a lexical or sentiment element u by a feature vector $\boldsymbol{u}$, where each lexical or sentiment element w that co-occurs with u in a review sentence contributes a feature to $\boldsymbol{u}$. Moreover, the value of the feature w in vector $\boldsymbol{u}$ is denoted by $f(\boldsymbol{u}, w)$. The vector $\boldsymbol{u}$ can be

TABLE 1
Positive (+) and negative (-) sentiment reviews in two different domains: *books* and *kitchen*.

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

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 elements	excellent*P, broad*P, survey*P, development*P, civilization*P, excellent+broad*P, broad+survey*P, survey+development*P, development+civilization*P

seen as a compact representation of the distribution of an element u over the set of elements that co-occur with u in the reviews. The Distributional hypothesis states that words that have similar distributions are semantically similar [16].

We compute $f(\boldsymbol{u}, w)$ as the pointwise mutual information between a lexical element u and a feature w as follows:

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

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 $N = \sum_{i=1}^{n} \sum_{j=1}^{m} c(i, j)$. Using pointwise mutual information to weight features has been shown to be useful in numerous tasks in natural language processing such as similarity measurement [17], word classification [18], and word clustering [19]. However, pointwise mutual information is known to be biased towards infrequent elements and features. We follow the discounting approach proposed by Pantel & Ravichandran [18] to overcome this bias.

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

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

The relatedness score $\tau(v, u)$ can be interpreted as the proportion of pmi-weighted features of the element $\boldsymbol{u}$ that are shared with element $\boldsymbol{v}$. Note that pointwise mutual information values can become negative in practice even after discounting for rare occurrences. To avoid considering negative pointwise mutual information values, we only consider positive weights in Equation 2. Note that relatedness is an asymmetric measure according 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.

In cross-domain sentiment classification the source and target domains are not symmetric. For example, consider the two domains shown in Table 1. Given the target domain (kitchen appliances) and the lexical element "energy saving", we must identify that it is similar in sentiment (positive) to a source domain (books) lexical element such as "well researched" and expand "energy saving" by "well researched", when we must classify a review in the target (kitchen appliances) domain. Conversely, let us assume that "energy saving" also appears in the books domain (e.g. a book about ecological systems that attempt to minimize the use of energy) but "well researched" does not appear in the kitchen appliances domain. Under such circumstances, we must not expand "well researched" by "energy saving" when we must classify a target (books) domain using a model trained on the source (kitchen appliances) domain reviews.

The relatedness measure defined in Equation 2 can be further explained as the co-occurrences of u that can be recalled using v according to the co-occurrence retrieval framework proposed by Weeds and Weir [20]. In Section 6.3, we empirically compare the proposed relatedness measure with several other popular relatedness measures in a cross-domain sentiment classification task.

We use the relatedness measure defined in Equation 2 to construct a *sentiment sensitive* thesaurus in which, for each lexical element u we list up lexical elements v that co-occur with v (i.e. $f(\boldsymbol{u}, v) > 0$) in the descending order of the relatedness values $\tau(v, u)$. For example, for the word *excellent* the sentiment sensitive thesaurus would list *awsome* and *delicious* as related words. In the remainder of the paper, we use the term *base entry* to refer to a lexical element u (e.g. *excellent* in the previous

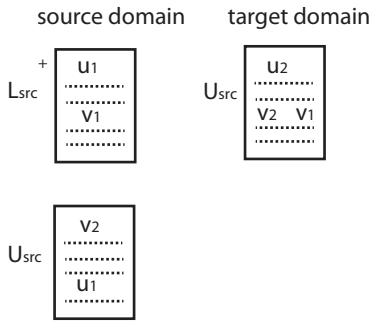

Fig. 1. Constructing feature vectors for two lexical elements u_1 and u_2 from a positive labeled source domain review L_{src}, two unlabeled reviews from source (U_{src}) and target (U_{tar}) domains. Vector $\boldsymbol{u}_1$ contains the sentiment element $v_1 * P$ and the lexical elements v_1, v_2. Vector $\boldsymbol{u}_2$ contains lexical elements v_1 and v_2. The relatedness, $\tau(u_1, u_2)$, between u_1 and u_2 is given by Equation 2.

example), for which its related lexical elements v (e.g. *awsome* and *delicious* in the previous example) are listed in the thesaurus. Moreover, the related words v of u are referred to as the *neighbors* of u.

As shown graphically in Figure 1, 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. This is an important fact that differentiates our sentiment-sensitive thesaurus from other distributional thesauri which do not consider sentiment information. For example, let us assume that the feature vector representing the word *excellent* contains both the lexical element *cooking* (extracted from an unlabeled review) and the sentiment element *spicy*P* (extracted from a positively labeled review). When computing the relatedness between *excellent* and another word (e.g. *delicious*) using Equation 2, features created from both labeled and unlabeled reviews will be used, thereby making the relatedness scores sensitive to sentiment.

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 elements, 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.

To construct the sentiment sensitive thesaurus, we must compute pairwise relatedness values using Equation 2 for numerous lexical elements. Moreover, to compute the pointwise mutual information values in feature vectors, we must store the co-occurrence information between numerous lexical and sentiment elements. By using a sparse matrix format and approximate vector similarity computation techniques [21], we can efficiently create a thesaurus from a large set of reviews. In particular, by using approximate vector similarity computation techniques we can avoid computing relatedness values between lexical elements that are likely to have very small relatedness scores thus are unlikely to become neighbors of a given base entry.

5 Feature Expansion

A fundamental problem in cross-domain sentiment classification is that features that appear in the source domains do not always appear in the target domain. Therefore, even if we train a classifier using labeled data from the source domains, the trained model cannot be readily used to classify test instances in the target domain. To overcome this problem, we propose a *feature expansion* method where we augment a feature vector with additional related features selected from the sentiment-sensitive thesaurus created in Section 4. In this section, we describe our feature expansion method.

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 $\text{score}(u_i, \boldsymbol{d})$ for each base entry in the thesaurus as follows:

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

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 $\text{score}(u_i, \boldsymbol{d})$ 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 base 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, $\text{score}(v_d^r, d)$, 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. On the other hand, the expanded features must have lower feature values compared to that of the original features in particular feature vector. We have set the feature values for the original features to their frequency in a review. Because Amazon product reviews are short, most features occur only once in a review. By using the inverse rank as the feature value for expanded features, we only take into account the relative ranking of base entries and at the same time assign feature values lower than that for the original features.

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 visualize 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. By adjusting the value of k, the number of base entries used for expanding a review, one can change the size of this latent space onto which the feature vectors are mapped (an alternative would be to select base entries with scores greater than some threshold value).

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. Once a binary classifier is trained, we can use it to predict the sentiment of a target domain review. We use the above-mentioned feature expansion method coupled with the sentiment-sensitive thesaurus to expand feature vectors at test time for the target domain as well.

TABLE 3
Number of reviews in the benchmark dataset.

Domain	positive	negative	unlabeled
kitchen	1000	1000	16746
DVDs	1000	1000	34377
electronics	1000	1000	13116
books	1000	1000	5947

6 Experiments

6.1 Dataset

We use the cross-domain sentiment classification dataset[1] prepared by Blitzer et al. [7] to compare the proposed method against previous work on cross-domain sentiment classification. This dataset consists of Amazon product reviews for four different product types: books, DVDs, electronics and kitchen appliances. Each review is assigned with a rating (0-5 stars), a reviewer name and location, a product name, a review title and date, and the review text. Reviews with rating > 3 are labeled as positive, whereas those with rating < 3 are labeled as negative. The overall structure of this benchmark dataset is shown in Table 6.1. For each domain, there are 1000 positive and 1000 negative examples, the same balanced composition as the polarity dataset constructed by Pang et al. [1]. The dataset also contains some unlabeled reviews for the four domains. This benchmark dataset has been used in much previous work on cross-domain sentiment classification and by evaluating on it we can directly compare the proposed method against existing approaches.

Following previous work, we randomly select 800 positive and 800 negative labeled reviews from each domain as training instances (total number of training instances are $1600 \times 4 = 6400$), and the remainder is used for testing (total number of test instances are $400 \times 4 = 1600$). 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. We create a sentiment sensitive thesaurus using labeled data from the source domain and unlabeled data from source and target domains as described in Section 4. We then use this thesaurus to expand the labeled feature vectors (train instances) from the source domains and train an L1 regularized logistic regression-based binary classifier (Classias) [2]. L1 regularization is shown to produce a sparse model, where most irrelevant features are assigned a zero weight [22].

1. http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
2. http://www.chokkan.org/software/classias/

TABLE 4
The effect of using a sentiment sensitive thesaurus for cross-domain sentiment classification.

Method	kitchen	DVDs	electronics	books
No Adapt	0.7261	0.6897	0.7053	0.6272
NSST	0.7750	0.7350	0.7550	0.7146
Proposed (SST)	**0.8518**	**0.7826**	**0.8386**	**0.7632**
In-Domain	*0.8770*	*0.8240*	*0.8440*	*0.8040*

This enables us to select useful features for classification in a systematic way without having to preselect features using heuristic approaches. In our preliminary experiments, we observed that the classification accuracy on two development target domains did not vary significantly with different L1 regularization parameter values. Therefore, we set the L1 regularization parameter to 1, which is the default setting in Classias, for all experiments described in this paper. Next, we use the trained classifier to classify reviews in the target domain. The thesaurus is again used to expand feature vectors from the target domain. This procedure is repeated for each domain in Table 6.1.

The above mentioned procedure creates four thesauri (each thesaurus is created by excluding labeled training data for a particular target domain). For example, from the three domains DVDs, electronics and books, we generate $53,586$ lexical elements and $62,744$ sentiment elements to create a thesaurus that is used to adapt a classifier trained on those three domains to the kitchen domain. Similar numbers of features are generated for the other domains as well. To avoid generating sparse and probably noisy features, we require that each feature occur in at least two different review sentences. We use classification accuracy on target domain as the evaluation metric. It is the fraction of the correctly classified target domain reviews from the total number of reviews in the target domain, and is defined as follows:

$$\text{Accuracy} = \frac{\text{no. of correctly classified target reviews}}{\text{total no. of reviews in the target domain}}. \quad (4)$$

6.2 Cross-Domain Sentiment Classification

To evaluate the benefit of using a sentiment sensitive thesaurus for cross-domain sentiment classification, we compare the proposed method against three baseline methods in Table 4. Next, we describe the methods compared in Table 4.

- **No Adapt**: This 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 a target domain. This can be considered as a lower bound that does not perform domain adaptation.
- **NSST (Non-sentiment Sensitive Thesaurus)**: To evaluate the benefit of using sentiment features on our proposed method, we create a thesaurus only using lexical elements. Lexical elements can be derived from both labeled and unlabeled reviews whereas, sentiment elements can be derived only from labeled reviews. We did not use rating information in the source domain labeled data in this baseline. A thesaurus is created using those features and subsequently used for feature expansion. A binary classifier is trained using the expanded features.
- **Proposed (SST: sentiment sensitive thesaurus)**: This is the proposed method described in this paper. We use the sentiment sensitive thesaurus created using the procedure described in Section 4 and use the thesaurus for feature expansion in a binary classifier.
- **In-Domain**: In this method, we train a binary classifier using the labeled data from the target domain. This method provides an upper bound for the cross-domain sentiment analysis. This upper baseline demonstrates the classification accuracy we can hope to obtain if we had labeled data for the target domain. Note that this is not a cross-domain classification setting.

Table 4 shows the classification accuracy of the above-mentioned methods for each of the four domains in the benchmark dataset as the target domain. Moreover, for each domain we have shown in boldface the best cross-domain sentiment classification results. Note that the **In-Domain** baseline is not a cross-domain sentiment classification setting and acts as a upper bound. From the results in Table 4, we see that the **Proposed** (sentiment-sensitive thesaurus) returns the best cross-domain sentiment classification accuracy for all four domains. The analysis of variance (ANOVA) and Tukey's honestly significant differences (HSD) tests on the classification accuracies for the four domains show that our proposed method is statistically significantly better than both the **no thesaurus** and **non-sentiment sensitive thesaurus** baselines, at confidence level 0.05. This shows that using the sentiment sensitive thesaurus for feature expansion is useful for cross-domain sentiment classification.

6.3 Effect of Relatedness Measures

The choice of the relatedness measure is an important decision in a thesauri-based approach. Different relatedness measures will list different lexical elements as neighbors for a particular lexical element. Therefore, the set of expansion candidates will be directly influenced by the relatedness measure used to create the thesaurus. To study the effect of the relatedness measure on the performance of the proposed method, we construct four sentiment sensitive thesauri using four different relatedness measures. We then conduct feature expansion and training in the same manner as described in Section 5 with all four relatedness measures. We use the three domains at a time as the sources and the remaining

TABLE 5
Comparison of different relatedness measures.

Method	kitchen	DVDs	electronics	books	Overall
Cosine	0.8342	0.7826	0.8363	0.7657	0.8047
Lin [19]	0.8367	0.7826	0.8438	0.7632	0.8066
Proposed	0.8518	0.7826	0.8386	0.7632	**0.8091**
Reversed	0.8342	0.7852	0.8463	0.7632	0.8072

domain as the target in this experiment. The classification accuracies obtained using the different relatedness measures are shown in Table 5. Next, we describe the four relatedness measures compared in Table 5.

- **Cosine Similarity**: This is the cosine of the angle between the two vectors that represent two lexical elements u and v. Using the notation introduced in Section 4, it can be computed as follows:

$$\tau(v,u) = \frac{\sum_{w\in\Gamma(\boldsymbol{v})} f(\boldsymbol{u},w)}{||u||\,||v||}, \tag{5}$$

$$||v|| = \sqrt{\sum_{w\in\Gamma(\boldsymbol{v})} (f(\boldsymbol{v},w))^2}, \tag{6}$$

$$||u|| = \sqrt{\sum_{w\in\Gamma(\boldsymbol{u})>0\}} (f(\boldsymbol{u},w))^2}.$$

  Here, $\Gamma(\boldsymbol{v}) = \{x|f(\boldsymbol{v},x) > 0\}$, is the set of features x that have positive pmi values in the feature vector for the element $\boldsymbol{v}$. Cosine similarity is widely used as a measure of relatedness in numerous tasks in natural language processing [23].
- **Lin's Similarity Measure**: We use the similarity measure proposed by Lin [19] for clustering similar words. This measure has shown to outperform numerous other similarity measures for word clustering tasks. It is computed as follows:

$$\tau(v,u) = \frac{\sum_{w\in\Gamma(v)\cap\Gamma(u)}(f(\boldsymbol{v},w)+f(\boldsymbol{u},w))}{\sum_{w\in\Gamma(v)} f(\boldsymbol{v},w) + \sum_{w\in\Gamma(u)} f(\boldsymbol{u},w)}. \tag{7}$$

- **Proposed**: This is the relatedness measure proposed in this paper and is defined by Equation 2. Unlike the **Cosine Similarity** and **Lin's Similarity Measure**, this relatedness measure is asymmetric.
- **Reversed**: As a baseline that demonstrates the asymmetric nature of the relatedness measure proposed in Equation 2, we swap the two arguments u and v in Equation 2 to construct a baseline relatedness measure. Specifically, the reversed baseline is computed as follows:

$$\tau(v,u) = \frac{\sum_{w\in\{x|f(\boldsymbol{u},x)>0\}} f(\boldsymbol{v},w)}{\sum_{w\in\{x|f(\boldsymbol{v},x)>0\}} f(\boldsymbol{v},w)}. \tag{8}$$

  Note that this baseline assigns higher relatedness scores to expansion candidates u that are in frequent in user reviews, because the denominator of Equation 8 contains the sum of pointwise mutual information values for words that co-occur with u.

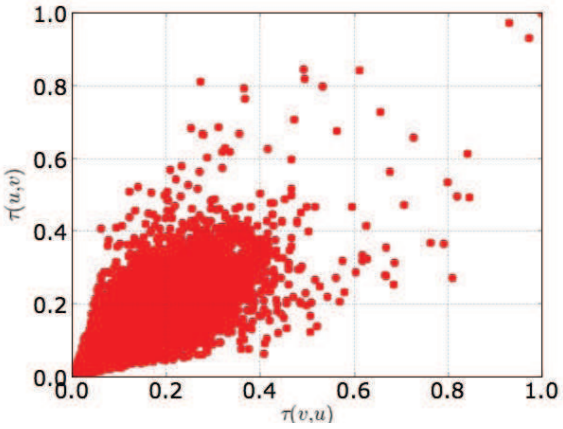

Fig. 2. Correlation between relatedness scores.

From Table 5 we see that the **Proposed** relatedness measure reports the highest overall classification accuracy followed by the **Reversed** baseline, **Lin's Similarity Measure**, and the **Cosine Similarity** in that order. However, it must be noted that the differences in performance among those relatedness measures are not statistically significant. This result implies that a wide-range of relatedness measures can be used to create a sentiment sensitive thesaurus to be used with the feature expansion method proposed in the paper. Further investigations into the insensitivity of the proposed method to the relatedness measures revealed three important reasons that we will discuss next.

First, recall that the proposed feature expansion method (Section 5) does not use the absolute value of relatedness scores, but only uses the relative rank among the expansion candidates. Therefore, two relatedness measures that produce different absolute scores can obtain similar performance if the relative rankings among expansion candidates are similar.

Second, as a posterior step to feature expansion we train a binary classifier with L1 regularization using source domain labeled data. Therefore, if we introduce any incorrect expansion candidates that do not properly reflect sentiment, those expansion candidates will be assigned zero weights. Consequently, invalid expansion candidates will be *pruned out* from the final model learnt by the binary classifier. However, it must be emphasized that although this posterior classifier training step can remove incorrect expansions, it *cannot* introduce the correct expansions. Therefore, it is vital to the performance of the proposed method that a relatedness measure identifies correct expansion candidates during the feature expansion step.

To study the degree of asymmetry in the relatedness measure proposed in Equation 2, and its effect on the performance of the proposed cross-domain sentiment classification method, we conduct the following experiment. For word pairs (u, v) in the sentiment sensitive

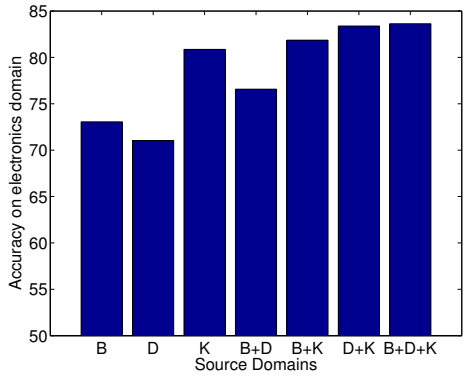

Fig. 3. Effect of using multiple source domains.

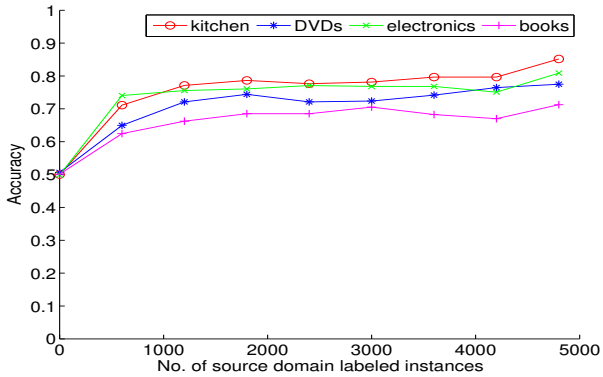

Fig. 4. Effect of source domain labeled data.

thesaurus, we plot the relatedness scores $\tau(u, v)$ against $\tau(v, u)$ as shown in Figure 2. There are $1,000,000$ such word pairs (data points) in Figure 2. From Figure 2, we see that $\tau(u, v)$ is highly correlated to $\tau(v, u)$. In fact the Pearson correlation coefficient for Figure 2 is as high as 0.8839 with a tight confidence interval of $[0.8835, 0.8844]$. This experimental result indicates that, although by definition Equation 2 is asymmetric, its level of asymmetry is very small in practice. Both the **Proposed** method and its **Reversed** baseline (Equation 8) reporting similar accuracy values in Table 5 further supports this finding. We consider this perceived low-level of asymmetry to be a third reason that explains the similar performance among symmetric and asymmetric relatedness measures compared in Table 5.

6.4 Effect of using Multiple Sources

In real-world cross-domain sentiment classification settings often we have more than one source domains at our disposal. Selecting the correct source domains to adapt to a given target domain is a challenging problem [24]. To study the effect of using multiple source domain in the proposed method, we select the electronics domain as the target and train a sentiment classifier using all possible combinations of the three source domains books (**B**), kitchen appliances (**K**), and DVDs (**D**). Note that we fix the total number of labeled training instances when we combine multiple domains as sources to avoid any performance gains simply because of the increased number of labeled instances as already explained in Section 6.1. Specifically, when using a single source domains we take 800 positive and 800 negative labeled reviews, when using two source domains we take 400 positive and 400 negative labeled reviews from each source domain, and when using all three source domains we take 266 positive and 266 negative labeled reviews. Moreover, we use all available unlabeled reviews from each source domain and the target domain.

Figure 3 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 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.

6.5 Effect of Source/Target Domain Dataset Sizes

The amount of training data that is required by a domain adaptation method to achieve a certain level of performance on a target domain is an important factor that determines the applicability of that method in real-world domain adaptation scenarios. There are three sources of training data in our task: source domain's labeled data (SL), source domain's unlabeled data (SU), and target domain's unlabeled data (TU).

To study the effect of SL, from each of the three source domains we select equal numbers of positive and negative instances and use those instances to train a classifier. Next, we evaluate the sentiment classification accuracy on the target domain. We repeat this process with each of the four domains as targets. Figure 4 shows the classification accuracy on the target domain against the total (positive and negative) number of labeled instances used from the three source domains. Without any labeled data, the classification accuracy is 0.5. Accuracy increases steadily upto 1200 labeled instances (200 positive and 200 negative instances from each domain) and then remains almost stable. The ability of the proposed method to reach its full performance with a small number of source domain labeled instances is particularly important when applying to domains with a few labeled instances.

To study the effect of SU and TU, we select the three domains books, electronics, and kitchen appliances as the source domains and DVDs as the target domain. In

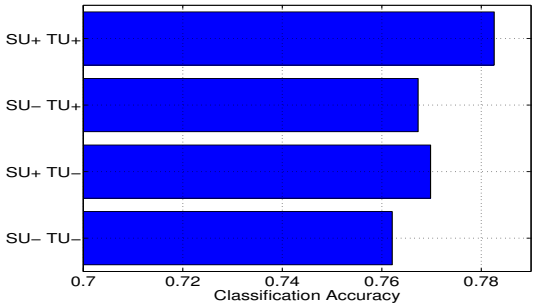

Fig. 5. Effect of source and target unlabeled data.

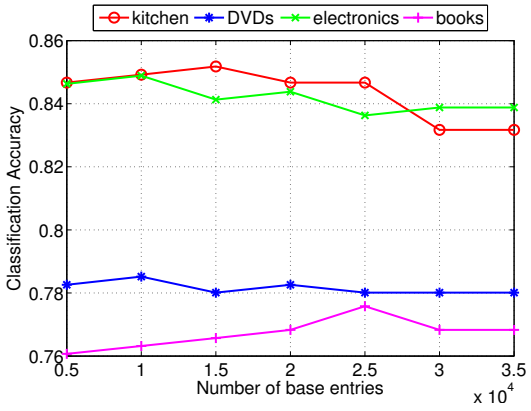

Fig. 6. Performance of the proposed method with the size of the sentiment sensitive thesaurus.

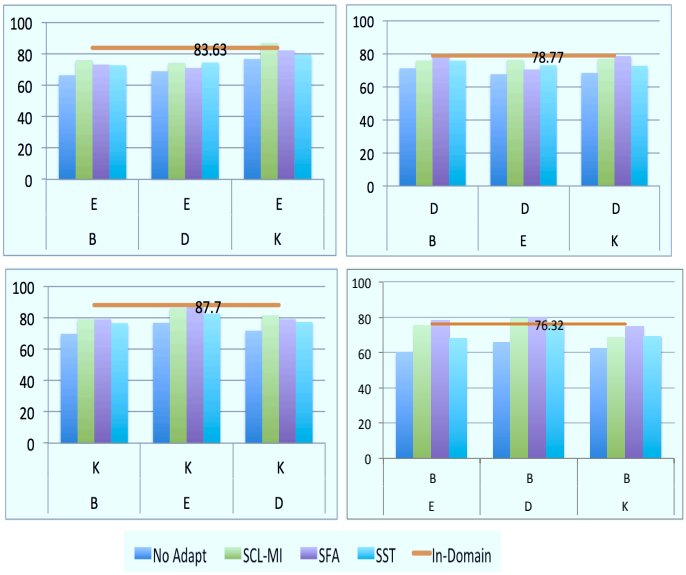

Fig. 7. Single Source Domain Adaptation.

Figure 5, we denote the presence and absence of a particular type of unlabeled data respectively by a + or − sign. For example, SU+ TU+ denotes the scenario where we have both source and target domains' unlabeled data whereas, SU- TU+ denotes the scenario where we only have target domain's unlabeled data. The amount of labeled data is kept fixed in this experiment. From Figure 5 we see that we obtain the best classification accuracy when we use both source and target unlabeled data (i.e. SU+ TU+). On the other hand, the worst performance is reported when we do not use any unlabeled data (i.e. SU- TU-). This result shows that unlabeled data plays an important role in our proposed method.

Figure 6 shows the cross-domain classification accuracy for the four target domains in the benchmark dataset. Overall we see that when we increase the size of the sentiment sensitive thesaurus (i.e. number of base entries) initially, the classification accuracy increases. This is because feature expansion enables us to reduce the mismatch between source and target domain features. However, when we further increase the size of the thesaurus, we see that the performs drops and then saturates. Note that when we increase the size of the thesaurus we will also increase the diversity of expansion candidates introduced by the feature expansion procedure. Note that although the total number of expansion candidates is held constant at 1000, we are selecting those 1000 candidates from a larger set of base entries when we increase the size of the sentiment sensitive thesaurus. However, the total number of source domain labeled instances remains constant even when we increase the size of the thesaurus. Therefore, we are unable to learn confident weights for all the expansion candidates, resulting in a saturation in performance.

6.6 Feature Analysis

To analyze the features learned by the proposed method we train the proposed method using kitchen, DVDs, and electronics as source domains. The proposed feature expansion method produces 137635 unique features for 4773 reviews. However, the L1 regularization produces a sparse model that contains only 1668 features by selecting the most discriminative features from the training instances. For three example features, Table 6 shows their model weights and top three expansions. Correct related features are found as expansion candidates by the proposed method. For example, *excellent* is expanded by bigram *invaluable+resource*, and *worst* is expanded by the bigram *absolute+junk*.

6.7 Comparison Against Previous Work

We compare our proposed method against two previously proposed cross-domain sentiment analysis methods. Next, we briefly describe those methods. They are described in detail in Section 8.

- **SCL-MI**: This is the structural correspondence learning (SCL) method proposed by Blitzer et al. [25]. This method utilizes both labeled and unlabeled data in the benchmark dataset. It selects pivots using the mutual information between a feature (unigrams or bigrams) and the domain label. Next,

TABLE 6
Some examples feature expansions produced by the proposed method.

Feature	weight	rank 1	rank 2	rank 3
excellent	2.27	invaluable+resource	definite+buy	delivery+prompt
great	1.24	definite+buy	bargain+bin	panini+sandwich
worst	−1.56	absolute+junk	hunk+junk	worthless+junk

linear classifiers are learnt to predict the existence of those pivots. The learnt weight vectors are arranged as rows in a matrix and singular value decomposition is performed to reduce the dimensionality of this matrix. Finally, this lower-dimensional matrix is used to project features to train a binary sentiment classifier.

- **SFA**: This is the spectral feature alignment (SFA) method proposed by Pan et al. [8]. Features are classified as to domain-specific or domain-independent using the mutual information between a feature and a domain label. Both unigrams and bigrams are considered as features to represent a review. Next, a bipartite graph is constructed between domain-specific and domain-independent features. An edge is formed between a domain-specific and a domain-independent feature in the graph if those two features co-occur in some feature vector. Spectral clustering is conducted to identify feature clusters. Finally, a binary classifier is trained using the feature clusters to classify positive and negative sentiment.

It must be emphasized that much previous work on cross-domain sentiment classification including the above-mentioned **SCL-MI** and **SFA** focus on using a single source domain to adapt to a particular target domain. Therefore, we first compare the proposed method (**SST**) against previous work in Figure 7, where we show the source domain on the top row and the target domain in the second row below the bar graphs. From Figure 7, we see that all methods outperform the **No Adapt** baseline consistently. Out of the 12 cases compared in Figure 7, **SCL-MI** reports the best accuracies among all cross-domain sentiment classification methods in 7 cases, whereas **SFA** reports the best accuracies in the remaining 5 cases.

However, as shown in Table 7, when we use multiple source domains we see that the proposed method outperforms both **SCL-MI** and **SFA** in all target domains, except for the books domain, where **SFA** outperforms the proposed method. This is because as we already saw in Figure 3, the accuracy of the proposed method improves when we combine multiple sources. However, the books domain has the lowest number of unlabeled reviews. Because the proposed method relies upon the availability of unlabeled data for the construction of a sentiment sensitive thesaurus, we believe that the lack of performance on books domain is a consequence of this. However, given that it is much cheaper to obtain unlabeled data for a target domain than labeled data, there is strong potential for improving the performance of the proposed method in this domain. Analysis of variance (ANOVA) and Tukey's honestly significant differences (HSD) tests show that the differences among **SCL**, **SFA** and the **Proposed** method are not statistically significant. Therefore, we conclude that the performance of the **Proposed** method is statistically comparable to that of **SCL-MI** and **SFA**.

TABLE 7
Comparison against previous work on multi-source cross-domain sentiment classification.

Method	kitchen	DVDs	electronics	books
No Adapt	0.7261	0.6897	0.7053	0.6272
SCL-MI [7]	0.8206	0.7630	0.7893	0.7456
SFA [8]	0.8148	0.7631	0.753	**0.7773**
Proposed (SST)	**0.8518**	**0.7826**	**0.8386**	0.7632
In-Domain	*0.8770*	*0.8240*	*0.8440*	*0.8040*

7 Comparison with SentiWordNet

We evaluate our sentiment sensitive thesaurus to group words that express similar sentiments. However, manual evaluation of a large thesaurus is costly. Instead, we compare the created sentiment sensitive thesaurus against SentiWordNet [26], a lexical database with polarity scores. SentiWordNet assigns each synset (a set of synonymous words for a particular sense of a word) in WordNet[3] with three types of scores in the range $[0, 1]$: *objective*, *positive*, and *negative*. If words in a synset express a positive sentiment then the synset will be assigned a high positive score, whereas if the words in a synset express a negative sentiment, then the synset will be asssigned a high negative score. If the words in a synset do not express any sentiment, then it will have a high objective score. SentiWordNet is created by automatically classifying the glosses associated with WordNet synsets using a set of eight ternary classifiers. It is freely available for research purposes[4].

We classify each non-objective (i.e. has a positive or a negative polarity score) word in SentiWordNet as positive, negative, or neutral as follows. If the degree of the positive polarity is greater than the degree of the negative polarity for a word, then it is classified as a positive word. If the degree of the negative polarity is greater than the positive polarity for a word, then it is classified as a negative word. If both the positive and

3. http://wordnet.princeton.edu/
4. http://sentiwordnet.isti.cnr.it/

negative polarity scores are equal, then it is classified as neutral. In SentiWordNet a particular word can have different polarity scores depending on its word sense. For example, the word *estimable* has an objective score of 1.0 (positive and negative scores of 0.0) for the sense *"may be computed or estimated"*, while the sense *"deserving of respect or high regard"* has a positive score of 0.75, negative score of 0.0 and an objective score of 0.25. Because the sentiment sensitive thesaurus does not have word sense information, it is not possible to know which sense of a word is listed in the thesaurus. Consequently, we classify a word to a particular class (positive, negative, or neutral) if at least one of the senses of the word can be classified to that class. After this classification, we have $18,829$ positive words, $21,043$ negative words, and $117,125$ neutral words.

We first create a single sentiment sensitive thesaurus using the training data for the four domains in the benchmark dataset (Table 6.1). Unlike the SentiWordNet, which is based on WordNet, a general purpose English lexicon, the benchmark dataset contains reviews for a limited set of domains. Therefore, we cannot expect to observe all the words that appear in the SentiWordNet in our training dataset. On the other hand, there are named entities such as product names that only occur in the training dataset but do not appear (thus do not have sentiment classifications) in SentiWordNet. Therefore, we only consider the words that appear in both SentiWordNet and in the sentiment sensitive thesaurus in our comparisons. For each word u listed as a base entry in the sentiment sensitive thesaurus, we generate pairs of words, (u, v), where v is listed as a neighbor for u in the thesaurus. We then check whether both u and v appear in positive, negative, or neutral word sets selected from the SentiWordNet as described above. If the proposed sentiment sensitive thesaurus can accurately group words that express similar sentiments, then we would expect a high percentage of word pairs (u, v) to be classified into one of those three classes as opposed to u and v belonging to different classes. In Table 8, we show the proportion of the correctly classified word pairs to the total number of word pairs in each class.

We compare the proposed sentiment sensitive thesaurus against two baselines in Table 8. The baseline sentiment **non-sensitive** uses a thesaurus created from the exact training dataset without using the sentiment elements as described in Section 4. This baseline demonstrates the effect of using sentiment elements in distributional similarity computations. The baseline **Lin** uses the distributional thesaurus created by Lin [19]. Note that Lin's thesaurus is a widely used distributional thesaurus that is constructed from a large web crawl. Compared to the sentiment sensitive thesaurus which has $723,621$ ($30,684$ unique words), and sentiment non-sensitive thesaurus which has $729,630$ words ($30,684$ unique words), the Lin's thesaurus has $8,846,513$ words ($108,556$ unique words). For each base entry, on average, there are twice as many related words listed in the Lin's thesaurus. **Lin** baseline demonstrates the ability of a general purpose large scale distributional thesaurus to group words that express similar sentiment together. From Table 8, we see that the proposed **sentiment sensitive** thesaurus shows the highest overall agreement (0.84) with SentiWordNet. Both **non-sensitive** and **Lin** baselines do not use sentiment related information in creating the thesaurus. The slight gain in overall accuracy for the **Lin** baseline over the **non-sensitive** baseline can be attributable its comparatively larger size.

TABLE 8
Comparison with SentiWordNet.

Thesaurus	positive	negative	neutral	overall
sentiment sensitive	0.19	0.11	0.54	0.84
non-sensitive	0.11	0.08	0.55	0.74
Lin	0.13	0.14	0.50	0.78

8 Related Work

Sentiment classification systems can be broadly categorized into single-domain [1], [2], [27]–[30] and cross-domain [7], [8] classifiers based upon the domains from which they are trained on and subsequently applied to. On another axis, sentiment classifiers can be categorized depending on whether they classify sentiment at word level [31], [32], sentence level [33], or document level [1], [2]. Our method performs cross-domain sentiment classification at document level.

In single-domain sentiment classification, a classifier is trained using labeled data annotated from the domain in which it will be applied. Turney [2] measures the co-occurrences between a word and a set of manually selected positive words (e.g. *good*, *nice*, *excellent*, etc.) and negative words (e.g. *bad*, *nasty*, *poor*, etc.) using pointwise mutual information to compute the sentiment of a word. Kanayama and Nasukawa [29] proposed an approach to build a domain-oriented sentiment lexicon to identify the words that express a particular sentiment in a given domain. By construction, a domain specific lexicon considers sentiment orientation of words in a particular domain. Therefore, their method cannot be readily applied to classify sentiment in a different domain.

Compared to single-domain sentiment classification, which has been studied extensively in previous work [3], cross-domain sentiment classification has only recently received attention with the advancement in the field of domain adaptation [34]–[36]. Aue and Gammon [37] reports a number of empirical tests on domain adaptation of sentiment classifiers. They use an ensemble of nine classifiers to train a sentiment classifier. However, most of these tests were unable to outperform a simple baseline classifier that is trained using all labeled data for all domains. They acknowledge the challenges involved in cross-domain sentiment classification and suggest the

possibilities of using unlabeled data to improve performance.

Blitzer et al. [7] propose the structural correspondence learning (SCL) algorithm to train a cross-domain sentiment classifier. SCL is motivated by the alternating structural optimization (ASO), a multi-task learning algorithm, proposed by Ando and Zhang [38]. Given labeled data from a source domain and unlabeled data from both source and target domains, SCL chooses a set of *pivot features* which occur frequently in both source and target domains. Next, linear predictors are trained to predict the occurrences of those pivot features. Positive training instances for a particular pivot feature are automatically generated by removing the corresponding pivot feature in feature vectors. Feature vectors that do not contain a particular pivot feature are considered as negative training instances for the task of learning a predictor for that pivot feature. It is noteworthy that this approach does not require any manually labeled feature vectors for learning the pivot feature predictors. For each pivot feature, a linear weight vector is computed and the set of weight vectors for all the pivot features under consideration are arranged in a matrix. Next, singular value decomposition (SVD) is performed on this weight matrix to construct a lower-dimensional feature space. Each feature vector is then mapped to a lower dimensional representation by multiplying with the computed matrix. Finally, each original feature vector is augmented with its lower dimensional representation to form a new (extended) feature vector. A binary classifier is trained using labeled reviews (positive and negative sentiment labels) using this new set of feature vectors. In the SCL-MI approach, a variant of the SCL approach, mutual information between a feature and the source label is used to select pivot features instead of the co-occurrence frequency. However, in practice it is hard to construct a reasonable number of auxiliary tasks from data, which might limit the transfer ability of SCL for cross-domain sentiment classification. Moreover, the 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 select a subset of features during training using L1 regularization. Moreover, we do not require SVD, cubic in time complexity, which can be computationally costly for large datasets.

Pan et al. [8] propose structural feature alignment (SFA) to find an alignment between domain specific and domain independent features. Mutual information of a feature with domain labels is used to classify domain-specific and domain-independent features. Next, a bipartite graph is constructed between domain-specific and domain-independent features. If a particular domain-specific features co-occurs with a domain-independent feature in some feature vector, then an edge is formed between the two features in the bipartite graph. Next, spectral clustering is performed on the bipartite graph that represents the two sets of features. However, not all words can be cleanly classified into domain specific or domain independent. Moreover, this classification is central to SFA and it is conducted as the first step in this method even before any classifier is trained. On the other hand, the thesaurus created by our method lets a particular lexical entry to be listed as related for multiple base entries. Moreover, we expand each feature vector individually and do not require any clustering. Consequently, as shown in Section 6, our method outperforms both SCL and SFA on a benchmark dataset of Amazon product reviews. Furthermore, unlike SCL and SFA, which consider a single source domain, our method can efficiently adapt from multiple source domains.

Domain adaptation in general has been studied in various other tasks such as part of speech tagging [39], named entity recognition [40], noun phrase chunking [34] and dependency parsing [41]. Domain adaptation methods can be broadly classified into fully supervised and semi-supervised adaptation [34]. In the fully supervised scenario we have labeled data for the source domain and also invest on labeling a few instances in the target domain. On the other hand, the semi-supervised version of domain adaptation does not assume the availability of labeled data from the target domain, but attempts to utilize a large set of unlabeled data selected from the target domain. Our proposed method falls under the semi-supervised domain adaptation category under this classification. Recently there has been some work on the theoretical aspects of domain adaptation [35], [36], [42].

9 Conclusion

We proposed a cross-domain sentiment classifier using an automatically extracted sentiment sensitive thesaurus. To overcome the feature mis-match problem in cross-domain sentiment classification, we use labeled data from multiple source domains and unlabeled data from source and target domains to compute the relatedness of features and construct a sentiment sensitive thesaurus. We then use the created thesaurus to expand feature vectors during train and test times for a binary classifier. A relevant subset of the features is selected using L1 regularization. The proposed method significantly outperforms several baselines and reports results that are comparable with previously proposed cross-domain sentiment classification methods on a benchmark dataset. Moreover, our comparisons against the SentiWordNet show that the created sentiment-sensitive thesaurus accurately groups words that express similar sentiments. In future, we plan to generalize the proposed method to solve other types of domain adaptation tasks.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *EMNLP 2002*, 2002, pp. 79–86.

[2] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," in *ACL 2002*, 2002, pp. 417–424.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[4] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *WWW 2009*, 2009, pp. 131–140.

[5] T.-K. Fan and C.-H. Chang, "Sentiment-oriented contextual advertising," *Knowledge and Information Systems*, vol. 23, no. 3, pp. 321–344, 2010.

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD 2004*, 2004, pp. 168–177.

[7] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL 2007*, 2007, pp. 440–447.

[8] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *WWW 2010*, 2010.

[9] H. Fang, "A re-examination of query expansion using lexical resources," in *ACL 2008*, 2008, pp. 139–147.

[10] G. Salton and C. Buckley, *Introduction to Modern Information Retreival*. McGraw-Hill Book Company, 1983.

[11] D. Shen, J. Wu, B. Cao, J.-T. Sun, Q. Yang, Z. Chen, and Y. Li, "Exploiting term relationship to boost text classification," in *CIKM'09*, 2009, pp. 1637 – 1640.

[12] T. Briscoe, J. Carroll, and R. Watson, "The second release of the rasp system," in *COLING/ACL 2006 Interactive Presentation Sessions*, 2006.

[13] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *ECML 1998*, 1998, pp. 137–142.

[14] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *ACL 1997*, 1997, pp. 174–181.

[15] J. M. Wiebe, "Learning subjective adjective from corpora," in *AAAI 2000*, 2000, pp. 735–740.

[16] Z. Harris, "Distributional structure," *Word*, vol. 10, pp. 146–162, 1954.

[17] P. Turney, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, 2006.

[18] P. Pantel and D. Ravichandran, "Automatically labeling semantic classes," in *NAACL-HLT'04*, 2004, pp. 321 – 328.

[19] D. Lin, "Automatic retrieval and clustering of similar words," in *ACL 1998*, 1998, pp. 768–774.

[20] J. Weeds and D. Weir, "Co-occurrence retrieval: A flexible framework for lexical distributional similarity," *Computational Linguistics*, vol. 31, no. 4, pp. 439–475, 2006.

[21] S. Sarawagi and A. Kirpal, "Efficient set joins on similarity predicates," in *SIGMOD '04*, 2004, pp. 743–754.

[22] A. Y. Ng, "Feature selection, l1 vs. l2 regularization, and rotational invariance," in *ICML 2004*, 2004.

[23] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 2002.

[24] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, "To transfer or not to transfer," in *NIPS 2005 Workshop on Transfer Learning*, 2005.

[25] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *EMNLP 2006*, 2006.

[26] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *LREC 2006*, 2006, pp. 417–422.

[27] G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding domain sentiment lexicon through double propagation," in *IJCAI 2009*, 2009, pp. 1199–1204.

[28] N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of html documents," in *EMNLP 2007*, 2007, pp. 1075–1083.

[29] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *EMNLP 2006*, 2006, pp. 355–363.

[30] H. Takamura, T. Inui, and M. Okumura, "Extracting semantic orientation of words using spin model," in *ACL 2005*, 2005, pp. 133–140.

[31] ——, "Extracting semantic orientation of phrases from dictionary," in *NAACL 2007*, 2007, pp. 292–299.

[32] E. Breck, Y. Choi, and C. Cardie, "Identifying expressions of opinion in context," in *IJCAI 2007*, 2007.

[33] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *EMNLP 2003*, 2003, pp. 129–136.

[34] H. Daumé III, "Frustratingly easy domain adaptation," in *ACL 2007*, 2007, pp. 256–263.

[35] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *NIPS 2008*, 2008.

[36] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, pp. 151–175, 2009.

[37] A. Aue and M. Gamon, "Customizing sentiment classifiers to new domains: a case study," Microsoft Research, Tech. Rep., 2005.

[38] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[39] K. Yoshida, Y. Tsuruoka, Y. Miyao, and J. Tsujii, "Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers," in *IJCAI 2007*, 2007, pp. 1783–1788.

[40] H. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su, "Domain adaptation with latent semantic association for named entity recognition," in *NAACL 2009*, 2009, pp. 281–289.

[41] M. Dredze, J. Blitzer, P. P. Talukdar, K. Ganchev, J. V. Graca, and F. Pereira, "Frustratingly hard domain adaptation for dependency parsing," in *CoNLL 2007*, 2007, pp. 1051–1055.

[42] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *NIPS 2006*, 2006.

Danushka Bollegala received his PhD from the University of Tokyo. He is an assistant professor (senior lecturer) at the University of Tokyo, working on natural language processing, machine learning, and data mining. He is a member of IEEE and ACL.

David Weir received his PhD from University of Pennsylvania. He is a Reader at University of Sussex. He has worked in numerous fields within natural language processing such as distributional semantics and parsing. He is a member of ACL.

John Carroll received his PhD from University of Cambridge. He is a professor at the University of Sussex, working on natural language parsing, automatic lexical acquisition, and application systems based on language processing. He is a member of ACL.