Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification

John Blitzer **Mark Dredze** **Fernando Pereira**
Department of Computer and Information Science
University of Pennsylvania
{blitzer|mdredze|pereria@cis.upenn.edu}

Abstract

Automatic sentiment classification has been extensively studied and applied in recent years. However, sentiment is expressed differently in different domains, and annotating corpora for every possible domain of interest is impractical. We investigate domain adaptation for sentiment classifiers, focusing on online reviews for different types of products. First, we extend to sentiment classification the recently-proposed structural correspondence learning (SCL) algorithm, reducing the relative error due to adaptation between domains by an average of 30% over the original SCL algorithm and 46% over a supervised baseline. Second, we identify a measure of domain similarity that correlates well with the potential for adaptation of a classifier from one domain to another. This measure could for instance be used to select a small set of domains to annotate whose trained classifiers would transfer well to many other domains.

1 Introduction

Sentiment detection and classification has received considerable attention recently (Pang et al., 2002; Turney, 2002; Goldberg and Zhu, 2004). While movie reviews have been the most studied domain, sentiment analysis has extended to a number of new domains, ranging from stock message boards to congressional floor debates (Das and Chen, 2001; Thomas et al., 2006). Research results have been deployed industrially in systems that gauge market reaction and summarize opinion from Web pages, discussion boards, and blogs.

With such widely-varying domains, researchers and engineers who build sentiment classification systems need to collect and curate data for each new domain they encounter. Even in the case of market analysis, if automatic sentiment classification were to be used across a wide range of domains, the effort to annotate corpora for each domain may become prohibitive, especially since product features change over time. We envision a scenario in which developers annotate corpora for a small number of domains, train classifiers on those corpora, and then apply them to other similar corpora. However, this approach raises two important questions. First, it is well known that trained classifiers lose accuracy when the test data distribution is significantly different from the training data distribution [1]. Second, it is not clear which notion of domain similarity should be used to select domains to annotate that would be good proxies for many other domains.

We propose solutions to these two questions and evaluate them on a corpus of reviews for four different types of products from Amazon: books, DVDs, electronics, and kitchen appliances[2]. First, we show how to extend the recently proposed structural cor-

[1] For surveys of recent research on domain adaptation, see the ICML 2006 Workshop on Structural Knowledge Transfer for Machine Learning (http://gameairesearch.uta.edu/) and the NIPS 2006 Workshop on Learning when test and training inputs have different distribution (http://ida.first.fraunhofer.de/projects/different06/)

[2] The dataset will be made available by the authors at publication time.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 440–447,
Prague, Czech Republic, June 2007.

respondence learning (SCL) domain adaptation algorithm (Blitzer et al., 2006) for use in sentiment classification. A key step in SCL is the selection of *pivot features* that are used to link the source and target domains. We suggest selecting pivots based not only on their common frequency but also according to their mutual information with the source labels. For data as diverse as product reviews, SCL can sometimes misalign features, resulting in degradation when we adapt between domains. In our second extension we show how to correct misalignments using a very small number of labeled instances.

Second, we evaluate the $\mathcal{A}$-distance (Ben-David et al., 2006) between domains as measure of the loss due to adaptation from one to the other. The $\mathcal{A}$-distance can be measured from unlabeled data, and it was designed to take into account only divergences which affect classification accuracy. We show that it correlates well with adaptation loss, indicating that we can use the $\mathcal{A}$-distance to select a subset of domains to label as sources.

In the next section we briefly review SCL and introduce our new pivot selection method. Section 3 describes datasets and experimental method. Section 4 gives results for SCL and the mutual information method for selecting pivot features. Section 5 shows how to correct feature misalignments using a small amount of labeled target domain data. Section 6 motivates the $\mathcal{A}$-distance and shows that it correlates well with adaptability. We discuss related work in Section 7 and conclude in Section 8.

2 Structural Correspondence Learning

Before reviewing SCL, we give a brief illustrative example. Suppose that we are adapting from reviews of computers to reviews of cell phones. While many of the features of a good cell phone review are the same as a computer review – the words "excellent" and "awful" for example – many words are totally new, like "reception". At the same time, many features which were useful for computers, such as "dual-core" are no longer useful for cell phones.

Our key intuition is that even when "good-quality reception" and "fast dual-core" are completely distinct for each domain, if they both have high correlation with "excellent" and low correlation with "awful" on *unlabeled* data, then we can tentatively align them. After learning a classifier for computer reviews, when we see a cell-phone feature like "good-quality reception", we know it should behave in a roughly similar manner to "fast dual-core".

2.1 Algorithm Overview

Given labeled data from a source domain and unlabeled data from both source and target domains, SCL first chooses a set of m pivot features which occur frequently in both domains. Then, it models the correlations between the pivot features and all other features by training linear pivot predictors to predict occurrences of each pivot in the unlabeled data from both domains (Ando and Zhang, 2005; Blitzer et al., 2006). The ℓth pivot predictor is characterized by its weight vector $\mathbf{w}_\ell$; positive entries in that weight vector mean that a non-pivot feature (like "fast dual-core") is highly correlated with the corresponding pivot (like "excellent").

The pivot predictor column weight vectors can be arranged into a matrix $W = [\mathbf{w}_\ell]_{\ell=1}^{n}$. Let $\theta \in \mathbb{R}^{k \times d}$ be the top k left singular vectors of W (here d indicates the total number of features). These vectors are the principal predictors for our weight space. If we chose our pivot features well, then we expect these principal predictors to discriminate among positive and negative words in both domains.

At training and test time, suppose we observe a feature vector $\mathbf{x}$. We apply the projection $\theta\mathbf{x}$ to obtain k new real-valued features. Now we learn a predictor for the augmented instance $\langle \mathbf{x}, \theta\mathbf{x} \rangle$. If θ contains meaningful correspondences, then the predictor which uses θ will perform well in both source and target domains.

2.2 Selecting Pivots with Mutual Information

The efficacy of SCL depends on the choice of pivot features. For the part of speech tagging problem studied by Blitzer et al. (2006), frequently-occurring words in both domains were good choices, since they often correspond to function words such as prepositions and determiners, which are good indicators of parts of speech. This is not the case for sentiment classification, however. Therefore, we require that pivot features also be good predictors of the source label. Among those features, we then choose the ones with highest mutual information to the source label. Table 1 shows the set-symmetric

SCL, not SCL-MI	SCL-MI, not SCL
book one <num> so all	***a_must a_wonderful loved_it***
very about they like	***weak don't_waste awful***
good when	***highly_recommended and_easy***

Table 1: Top pivots selected by SCL, but not SCL-MI (left) and vice-versa (right)

differences between the two methods for pivot selection when adapting a classifier from books to kitchen appliances. We refer throughout the rest of this work to our method for selecting pivots as SCL-MI.

3 Dataset and Baseline

We constructed a new dataset for sentiment domain adaptation by selecting Amazon product reviews for four different product types: books, DVDs, electronics and kitchen appliances. Each review consists of 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 were labeled positive, those with rating < 3 were labeled negative, and the rest discarded because their polarity was ambiguous. After this conversion, we had 1000 positive and 1000 negative examples for each domain, the same balanced composition as the polarity dataset (Pang et al., 2002). In addition to the labeled data, we included between 3685 (DVDs) and 5945 (kitchen) instances of unlabeled data. The size of the unlabeled data was limited primarily by the number of reviews we could crawl and download from the Amazon website. Since we were able to obtain labels for all of the reviews, we also ensured that they were balanced between positive and negative examples, as well.

While the polarity dataset is a popular choice in the literature, we were unable to use it for our task. Our method requires many unlabeled reviews and despite a large number of IMDB reviews available online, the extensive curation requirements made preparing a large amount of data difficult [3].

For classification, we use linear predictors on unigram and bigram features, trained to minimize the Huber loss with stochastic gradient descent (Zhang, 2004). On the polarity dataset, this model matches the results reported by Pang et al. (2002). When we report results with SCL and SCL-MI, we require that pivots occur in more than five documents in each domain. We set k, the number of singular vectors of the weight matrix, to 50.

4 Experiments with SCL and SCL-MI

Each labeled dataset was split into a training set of 1600 instances and a test set of 400 instances. All the experiments use a classifier trained on the training set of one domain and tested on the test set of a possibly different domain. The baseline is a linear classifier trained without adaptation, while the gold standard is an in-domain classifier trained on the same domain as it is tested.

Figure 1 gives accuracies for all pairs of domain adaptation. The domains are ordered clockwise from the top left: books, DVDs, electronics, and kitchen. For each set of bars, the first letter is the source domain and the second letter is the target domain. The thick horizontal bars are the accuracies of the in-domain classifiers for these domains. Thus the first set of bars shows that the baseline achieves 72.8% accuracy adapting from DVDs to books. SCL-MI achieves 79.7% and the in-domain gold standard is 80.4%. We say that the *adaptation loss* for the baseline model is 7.6% and the adaptation loss for the SCL-MI model is 0.7%. The *relative reduction in error due to adaptation* of SCL-MI for this test is 90.8%.

We can observe from these results that there is a rough grouping of our domains. Books and DVDs are similar, as are kitchen appliances and electronics, but the two groups are different from one another. Adapting classifiers from books to DVDs, for instance, is easier than adapting them from books to kitchen appliances. We note that when transferring from kitchen to electronics, SCL-MI actually outperforms the in-domain classifier. This is possible since the unlabeled data may contain information that the in-domain classifier does not have access to.

At the beginning of Section 2 we gave examples of how features can change behavior across domains. The first type of behavior is when predictive features from the source domain are not predictive or do not appear in the target domain. The second is

[3]For a description of the construction of the polarity dataset, see `http://www.cs.cornell.edu/people/pabo/movie-review-data/`.

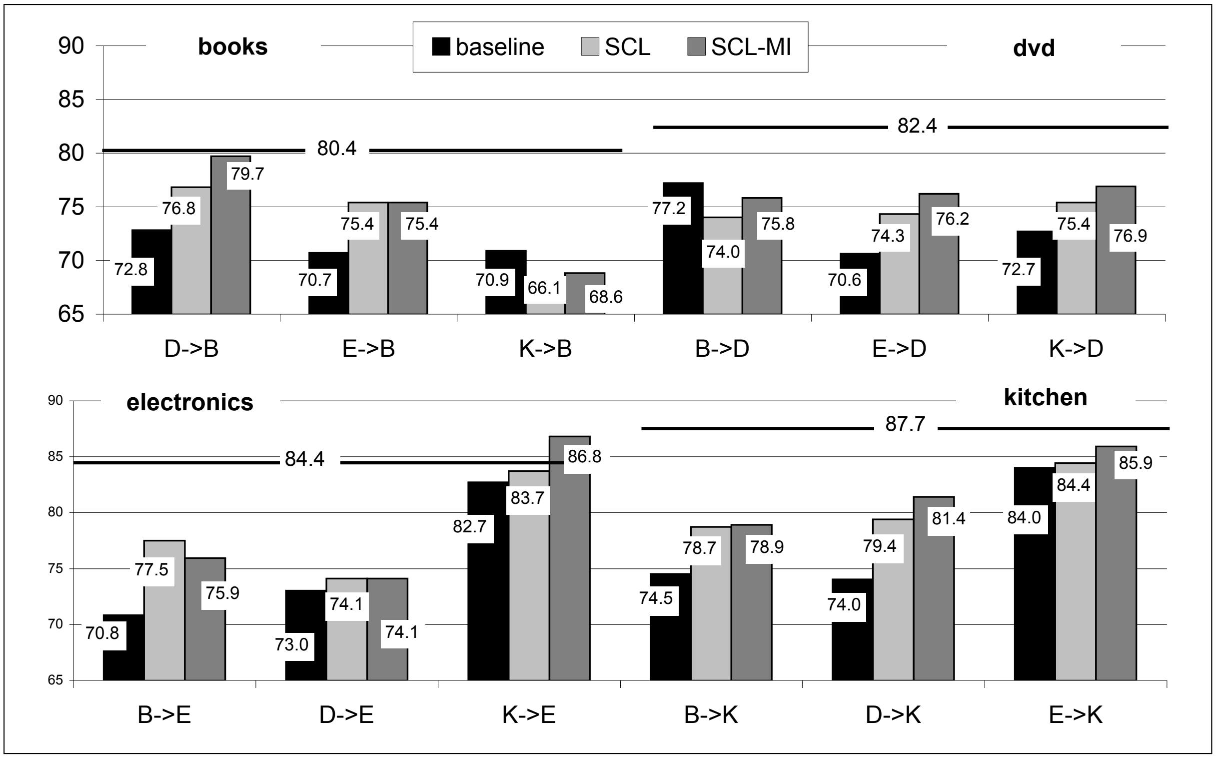

Figure 1: Accuracy results for domain adaptation between all pairs using SCL and SCL-MI. Thick black lines are the accuracies of in-domain classifiers.

domain\polarity	negative	positive
books	***plot <num>_pages predictable reading_this page_<num>***	***reader grisham engaging must_read fascinating***
kitchen	***the_plastic poorly_designed leaking awkward_to defective***	***excellent_product espresso are_perfect years_now a_breeze***

Table 2: Correspondences discovered by SCL for books and kitchen appliances. The top row shows features that only appear in books and the bottom features that only appear in kitchen appliances. The left and right columns show negative and positive features in correspondence, respectively.

when predictive features from the target domain do not appear in the source domain. To show how SCL deals with those domain mismatches, we look at the adaptation from book reviews to reviews of kitchen appliances. We selected the top 1000 most informative features in both domains. In both cases, between 85 and 90% of the informative features from one domain were not among the most informative of the other domain[4]. SCL addresses both of these issues simultaneously by aligning features from the two domains.

[4]There is a third type, features which are positive in one domain but negative in another, but they appear very infrequently in our datasets.

Table 2 illustrates one row of the projection matrix θ for adapting from books to kitchen appliances; the features on each row appear only in the corresponding domain. A supervised classifier trained on book reviews cannot assign weight to the kitchen features in the second row of table 2. In contrast, SCL assigns weight to these features indirectly through the projection matrix. When we observe the feature "predictable" with a negative book review, we update parameters corresponding to the entire projection, including the kitchen-specific features "poorly_designed" and "awkward_to".

While some rows of the projection matrix θ are useful for classification, SCL can also misalign features. This causes problems when a projection is discriminative in the source domain but not in the target. This is the case for adapting from kitchen appliances to books. Since the book domain is quite broad, many projections in books model topic distinctions such as between religious and political books. These projections, which are uninformative as to the target label, are put into correspondence with the fewer discriminating projections in the much narrower kitchen domain. When we adapt from kitchen to books, we assign weight to these uninformative projections, degrading target classification accuracy.

5 Correcting Misalignments

We now show how to use a small amount of target domain labeled data to learn to ignore misaligned projections from SCL-MI. Using the notation of Ando and Zhang (2005), we can write the supervised training objective of SCL on the source domain as

$$\min_{\mathbf{w},\mathbf{v}} \sum_i L\left(\mathbf{w}'\mathbf{x}_i + \mathbf{v}'\theta\mathbf{x}_i, y_i\right) + \lambda||\mathbf{w}||^2 + \mu||\mathbf{v}||^2 ,$$

where y is the label. The weight vector $\mathbf{w} \in \mathbb{R}^d$ weighs the original features, while $\mathbf{v} \in \mathbb{R}^k$ weighs the projected features. Ando and Zhang (2005) and Blitzer et al. (2006) suggest $\lambda = 10^{-4}, \mu = 0$, which we have used in our results so far.

Suppose now that we have trained source model weight vectors $\mathbf{w}_s$ and $\mathbf{v}_s$. A small amount of target domain data is probably insufficient to significantly change $\mathbf{w}$, but we can correct $\mathbf{v}$, which is much smaller. We augment each labeled target instance $\mathbf{x}_j$ with the label assigned by the source domain classifier (Florian et al., 2004; Blitzer et al., 2006). Then we solve

$$\min_{\mathbf{w},\mathbf{v}} \sum_j L\left(\mathbf{w}'\mathbf{x}_j + \mathbf{v}'\theta\mathbf{x}_j, y_j\right) + \lambda||\mathbf{w}||^2 + \mu||\mathbf{v} - \mathbf{v}_s||^2 .$$

Since we don't want to deviate significantly from the source parameters, we set $\lambda = \mu = 10^{-1}$.

Figure 2 shows the corrected SCL-MI model using 50 target domain labeled instances. We chose this number since we believe it to be a reasonable amount for a single engineer to label with minimal effort. For reasons of space, for each target domain we show adaptation from only the two domains on which SCL-MI performed the worst relative to the supervised baseline. For example, the book domain shows only results from electronics and kitchen, but not DVDs. As a baseline, we used the label of the source domain classifier as a feature in the target, but did not use any SCL features. We note that the baseline is very close to just using the source domain classifier, because with only 50 target domain instances we do not have enough data to relearn all of the parameters in $\mathbf{w}$. As we can see, though, relearning the 50 parameters in $\mathbf{v}$ is quite helpful. The corrected model *always* improves over the baseline for every possible transfer, including those not shown in the figure.

dom \ model	base	base +targ	scl	scl-mi	scl-mi +targ
books	8.9	9.0	7.4	5.8	**4.4**
dvd	8.9	8.9	7.8	6.1	**5.3**
electron	8.3	8.5	6.0	5.5	**4.8**
kitchen	10.2	9.9	7.0	5.6	**5.1**
average	9.1	9.1	7.1	5.8	**4.9**

Table 3: For each domain, we show the loss due to transfer for each method, averaged over all domains. The bottom row shows the average loss over all runs.

The idea of using the regularizer of a linear model to encourage the target parameters to be close to the source parameters has been used previously in domain adaptation. In particular, Chelba and Acero (2004) showed how this technique can be effective for capitalization adaptation. The major difference between our approach and theirs is that we only penalize deviation from the source parameters for the weights $\mathbf{v}$ of projected features, while they work with the weights of the original features only. For our small amount of labeled target data, attempting to penalize $\mathbf{w}$ using $\mathbf{w}_s$ performed no better than our baseline. Because we only need to learn to ignore projections that misalign features, we can make much better use of our labeled data by adapting only 50 parameters, rather than 200,000.

Table 3 summarizes the results of sections 4 and 5. Structural correspondence learning reduces the error due to transfer by 21%. Choosing pivots by mutual information allows us to further reduce the error to 36%. Finally, by adding 50 instances of target domain data and using this to correct the misaligned projections, we achieve an average relative

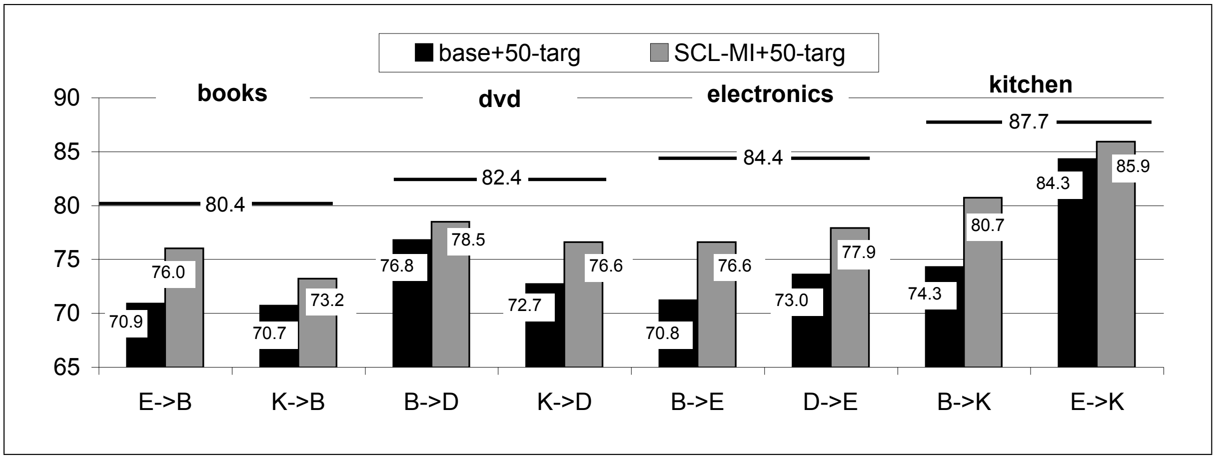

Figure 2: Accuracy results for domain adaptation with 50 labeled target domain instances.

reduction in error of 46%.

6 Measuring Adaptability

Sections 2-5 focused on how to adapt to a target domain when you had a labeled source dataset. We now take a step back to look at the problem of selecting source domain data to label. We study a setting where an engineer knows roughly her domains of interest but does not have any labeled data yet. In that case, she can ask the question "Which sources should I label to obtain the best performance over all my domains?" On our product domains, for example, if we are interested in classifying reviews of kitchen appliances, we know from sections 4-5 that it would be foolish to label reviews of books or DVDs rather than electronics. Here we show how to select source domains using only unlabeled data and the SCL representation.

6.1 The $\mathcal{A}$-distance

We propose to measure domain adaptability by using the divergence of two domains after the SCL projection. We can characterize domains by their induced distributions on instance space: the more different the domains, the more divergent the distributions. Here we make use of the $\mathcal{A}$-distance (Ben-David et al., 2006). The key intuition behind the $\mathcal{A}$-distance is that while two domains can differ in arbitrary ways, we are only interested in the differences that affect classification accuracy.

Let $\mathcal{A}$ be the family of subsets of $\mathbb{R}^k$ corresponding to characteristic functions of linear classifiers (sets on which a linear classifier returns positive value). Then the $\mathcal{A}$ distance between two probability distributions is

$$d_{\mathcal{A}}(\mathcal{D}, \mathcal{D}') = 2 \sup_{A \in \mathcal{A}} \left| \Pr_{\mathcal{D}} [A] - \Pr_{\mathcal{D}'} [A] \right| \, .$$

That is, we find the subset in $\mathcal{A}$ on which the distributions differ the most in the L_1 sense. Ben-David et al. (2006) show that computing the $\mathcal{A}$-distance for a finite sample is exactly the problem of minimizing the empirical risk of a classifier that discriminates between instances drawn from $\mathcal{D}$ and instances drawn from $\mathcal{D}'$. This is convenient for us, since it allows us to use classification machinery to compute the $\mathcal{A}$-distance.

6.2 Unlabeled Adaptability Measurements

We follow Ben-David et al. (2006) and use the Huber loss as a proxy for the $\mathcal{A}$-distance. Our procedure is as follows: Given two domains, we compute the SCL representation. Then we create a data set where each instance $\theta\mathbf{x}$ is labeled with the identity of the domain from which it came and train a linear classifier. For each pair of domains we compute the empirical average per-instance Huber loss, subtract it from 1, and multiply the result by 100. We refer to this quantity as the proxy $\mathcal{A}$-distance. When it is 100, the two domains are completely distinct. When it is 0, the two domains are indistinguishable using a linear classifier.

Figure 3 is a correlation plot between the proxy $\mathcal{A}$-distance and the adaptation error. Suppose we wanted to label two domains out of the four in such a

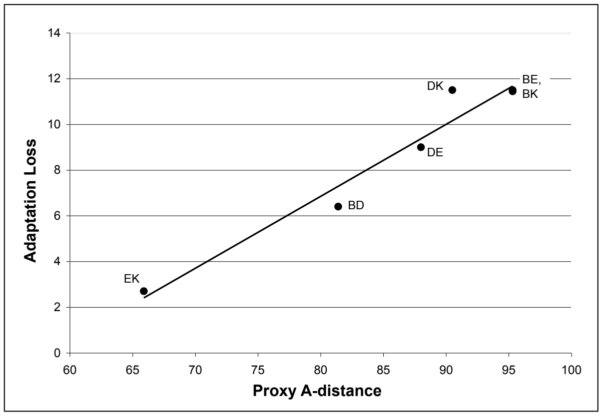

Figure 3: The proxy $\mathcal{A}$-distance between each domain pair plotted against the average adaptation loss of as measured by our baseline system. Each pair of domains is labeled by their first letters: EK indicates the pair electronics and kitchen.

way as to minimize our error on all the domains. Using the proxy $\mathcal{A}$-distance as a criterion, we observe that we would choose one domain from either books or DVDs, but not both, since then we would not be able to adequately cover electronics or kitchen appliances. Similarly we would also choose one domain from either electronics or kitchen appliances, but not both.

7 Related Work

Sentiment classification has advanced considerably since the work of Pang et al. (2002), which we use as our baseline. Thomas et al. (2006) use discourse structure present in congressional records to perform more accurate sentiment classification. Pang and Lee (2005) treat sentiment analysis as an ordinal ranking problem. In our work we only show improvement for the basic model, but all of these new techniques also make use of lexical features. Thus we believe that our adaptation methods could be also applied to those more refined models.

While work on domain adaptation for sentiment classifiers is sparse, it is worth noting that other researchers have investigated unsupervised and semisupervised methods for domain adaptation. The work most similar in spirit to ours that of Turney (2002). He used the difference in mutual information with two human-selected features (the words "excellent" and "poor") to score features in a completely unsupervised manner. Then he classified documents according to various functions of these mutual information scores. We stress that our method improves a supervised baseline. While we do not have a direct comparison, we note that Turney (2002) performs worse on movie reviews than on his other datasets, the same type of data as the polarity dataset.

We also note the work of Aue and Gamon (2005), who performed a number of empirical tests on domain adaptation of sentiment classifiers. Most of these tests were unsuccessful. We briefly note their results on combining a number of source domains. They observed that source domains closer to the target helped more. In preliminary experiments we confirmed these results. Adding more labeled data always helps, but diversifying training data does not. When classifying kitchen appliances, for any fixed amount of labeled data, it is always better to draw from electronics as a source than use some combination of all three other domains.

Domain adaptation alone is a generally well-studied area, and we cannot possibly hope to cover all of it here. As we noted in Section 5, we are able to significantly outperform basic structural correspondence learning (Blitzer et al., 2006). We also note that while Florian et al. (2004) and Blitzer et al. (2006) observe that including the label of a source classifier as a feature on small amounts of target data tends to improve over using either the source alone or the target alone, we did not observe that for our data. We believe the most important reason for this is that they explore structured prediction problems, where labels of surrounding words from the source classifier may be very informative, even if the current label is not. In contrast our simple binary prediction problem does not exhibit such behavior. This may also be the reason that the model of Chelba and Acero (2004) did not aid in adaptation.

Finally we note that while Blitzer et al. (2006) did combine SCL with labeled target domain data, they only compared using the label of SCL or non-SCL source classifiers as features, following the work of Florian et al. (2004). By only adapting the SCL-related part of the weight vector $\mathbf{v}$, we are able to make better use of our small amount of unlabeled data than these previous techniques.

8 Conclusion

Sentiment classification has seen a great deal of attention. Its application to many different domains of discourse makes it an ideal candidate for domain adaptation. This work addressed two important questions of domain adaptation. First, we showed that for a given source and target domain, we can significantly improve for sentiment classification the structural correspondence learning model of Blitzer et al. (2006). We chose pivot features using not only common frequency among domains but also mutual information with the source labels. We also showed how to correct structural correspondence misalignments by using a small amount of labeled target domain data.

Second, we provided a method for selecting those source domains most likely to adapt well to given target domains. The unsupervised $\mathcal{A}$-distance measure of divergence between domains correlates well with loss due to adaptation. Thus we can use the $\mathcal{A}$-distance to select source domains to label which will give low target domain error.

In the future, we wish to include some of the more recent advances in sentiment classification, as well as addressing the more realistic problem of ranking. We are also actively searching for a larger and more varied set of domains on which to test our techniques.

Acknowledgements

We thank Nikhil Dinesh for helpful advice throughout the course of this work. This material is based upon work partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD03001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the Department of Interior-National BusinessCenter (DOI-NBC).

References

Rie Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853.

Anthony Aue and Michael Gamon. 2005. Customizing sentiment classifiers to new domains: a case study. http://research.microsoft.com/ anthaue/.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Neural Information Processing Systems (NIPS)*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *EMNLP*.

Sanjiv Das and Mike Chen. 2001. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of Athe Asia Pacific Finance Association Annual Conference*.

R. Florian, H. Hassan, A.Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *of HLT-NAACL*.

Andrew Goldberg and Xiaojin Zhu. 2004. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of Association for Computational Linguistics*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of Empirical Methods in Natural Language Processing*.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of Association for Computational Linguistics*.

Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *International Conference on Machine Learning (ICML)*.