In Pursuit of an Efficient Multi-Domain Text Classification Algorithm

Evan Cox
Stanford University
evancox@stanford.edu

Marcelo Worsley
Stanford University
marcelo.worsley@stanford.edu

Abstract

Text classification remains an important part of natural language processing and information extraction research. Of particular interest are techniques that allow individuals to construct classifiers that have relevance for a variety of domains. This problem is particularly difficult because it is dependent on both inter-and intra-domain variance. In an attempt to address these challenges, researchers have begun to explore domain adaptation. In this paper we present two approaches for addressing this problem. We will show that generalizable features can be useful for doing classification in mixed domain datasets, and also show how establishing relationship mappings between words of different domains can help better understand domain adaptation.

1. Introduction

Text classification remains a salient topic in natural language processing communities. Applications of text classification are gradually expanding to an ever growing number of domains. However, many challenges still remain in constructing classifiers that can be effective across different domains as most models remains highly domain specific. Domain adaptive models become increasingly pertinent when one enters a scenario in which 1) sufficient domain specific information is not available or 2) the target domain data is not labeled. Finally, domain adaptation is further complicated by the highly varied effectiveness of "state-of-the-art" machine learning techniques for text classification.

In this paper we will focus on our efforts to construct models that can generalize from one domain to another. All of the work can be classified as sentiment analysis of product reviews. However, unlike traditional sentiment analysis, which looks to determine either positive or negative sentiment, our analyses were based on the common 5-star scale which helps express different levels of satisfaction.

This work was additionally motivated by the authors' previous work in this space. This previous work involved the development of sub-product category specific classifiers (eg. televisions would be a sub-category of electronics). The aftermath of this study was a series of classifiers that were successful at doing sub-category specific predictions, but performed poorly on doing category-wide predictions. Similarly, the authors also did previous work on sarcasm detection in texts. This study was somewhat effective at doing accurate predictions, but suffered from picking up non-substantive feature differentiation. This is to say that some of the most informative features had little semantic or syntactic meaning.

2. Related Work

In this section we outline some of the previous work related to text categorization and domain adaptation. Particular attention is devoted to techniques that helped inform our project implementation.

2.1. Text Categorization Techniques

Work on text classification has been quite extensive. As such, there are too many techniques to describe in this paper. Instead, we'll provide a brief overview of some of the leading techniques that are used.

2.1.1 Decision Trees

Decision trees are often heralded because of their ability to easily depict a specific classification process. In order to determine the most likely classification for a document, one need only progress through a series of decision nodes. Each node is intended to represent high entropy features, with the decision node indicating the predicted class.

2.1.2 Naïve Bayes

Naïve Bayes is often equated to the bag of words model. In principle, Naïve Bayes makes the assumption that each feature is independent of the others. As such, determining the probability of a given class based on the features present reduces to a fairly simple product. There are two primary variants of Naïve Bayes. Multivariate, which necessarily normalizes the vector space of each document by providing a binary indicator of whether or not a given feature is present; and multinomial, which produces a non-sparse matrix that takes into account the number of times different words occur within a document.

In the area of text categorization, the multinomial approach consistently yields more accurate models.

2.1.3 Maximum Entropy Markovization

Maximum entropy models eliminate the double counting that often plagues Naïve Bayes models. Furthermore, by looking to maximum entropy, the model can help obviate many of the problems associated with over-fitting (assuming one's features are well constructed). Additionally, when combined with Markovization, ones is able to effectively capture word context. This can help ensure good precision and recall.

2.1.4 Support Vector Machines

In many respects, SVM represents the state-of-the-art in text classification. By exploiting large margins, and realizing the training examples that exemplify the decision boundaries, SVM is normally able to produce effective classifiers. In the case of non-binary relationships, traditional SVM can be easily augmented to perform a series of pair-wise decision boundaries.

2.2. Domain Adaptation

Recent work in this area has seen the influx of a number of novel techniques. Here we quickly describe a few of these techniques as they helped inform the design of our techniques.

2.2.1 Structural Correspondence Learning

The fundamental goal of structural correspondence learning is to discover similar relationships between features in different domains. Blitzer provides an example of the word 'signal', in biomedical journals, behaving the same way as 'buyout' or 'investment' in the Wall Street Journal Corpus. These corresponding features are referred to as pivot features, and form the majority of the features in SCL. Studies have shown that they are an effective way for capturing similarities across different domains. Furthermore, they are not limited to text categorization applications, but are also useful for other types of classifications, including POS tagging.

2.2.2 Source, Target and General Models

The essence of this approach is to create three separate models. There is a model that is specific to the source domain, another that is specific to the target, and a third that is general to both domains. One way of thinking about this is as partition ones features or training example as specific to the target or source, or as general. By doing this, one can exploit those features that work best for each domain. However, this model is stagnated by its unrealistically large training time.

2.2.3 Feature Augmentation

Similar to 2.2.2, the idea behind feature augmentation is to develop a model that understands the relative impact that each features has on a given domain. It does this by increasing the feature space by a factor of 3. Doing this enables one to compute vector weights that indicate how large of an influence a given feature has on the target domain, source domain or the common domain. By doing this, Daumé and Marcu were able to use this technique to improve the accuracy of named-entity recognition, shallow parsing and part-of-speech tagging on a number of common datasets. In this work they also determined that target features tend to have twice as much predictive power as source features.

2.2.4 Adaptive Naïve Bayes

Like many of the other techniques described in this section, Adaptive Naïve Bayes wishes to identify generalizable features between the source domain and the target domain. This feature similarity is computed using Frequently Co-occuring Entropy which captures features that have both a high rate of co-occurrence and a similar probability of occurrence. In order to train this classifier Tan et al. use an adapting expectation maximization algorithm that adjusts the weights for the generalizable features from the source domain and all of the features from the target domain. The 'adaptive' part is embedded in the algorithms gradually increasing the weight attributed to features of the target domain. In practice this technique has met with mixed success with the authors suggesting the need for a superior similarity metric.

3. Data

Data was obtained from the Potts Corpora. While this corpora contains a number of product-reviews in different languages, we chose to only utilize the English product reviews from Tripadvisor and Amazon. Limiting the language to English was done to ensure that we could adequately comment on the semantic and syntactic structure of the texts. The data that we selected is partitioned into a set of reviews for each of the following category: books, electronics, hotels and restaurants. Additionally we used a set of mixed reviews from Amazon as an extra piece of testing material for our system.

3.1. Preprocessing Data

Data was initially provided in xml format. It was then processed using a series of python scripts to remove html tags. Libraries from the python Natural Language Toolkit were used to parse sentences, and subsequently tokenize words and punctuation. The final outcome of this pre-processing was a collection of files that were neatly

segmented into individual files based on the product domain and the 5-star rating.

This collection of product ratings and reviews was then randomly sampled to be used in the following approaches.

4. Approach

4.1. Text Classification via Extraction of Common N-gram Features

This approach was largely motivated by a desire to understand the types of features that are consistent across multiple domains. Furthermore, it looked to realize those features that have high predictive power across domains. Even though some research has shown that generalizable features by themselves may not be sufficient for building complete models, understanding which features generally indicate the degree of positive or negative sentiment would help validate the idea that there is some relative consistency in the way people review products in different domains.

4.1.1 Random Sampling of 5000 Product Reviews

5000 reviews were randomly selected from each dataset to be included in this analysis. Multiple trials were performed in order to ensure result consistency. Below is a table that describes the number of reviews, and the size of each vocabulary.

	Samples	**Vocabulary**
Books	4820	18860
Electronics	4992	12156
Hotels	4771	9390
Restaurants	4971	5440

4.1.2 Tokenization to produce N-grams

Each review was then processed to produce all possible n-grams. The resulting features included unigrams, bigrams and trigrams. These n-grams were then pruned to eliminate all n-grams that occurred in the dataset fewer than 100 times.

4.1.3 Selecting N-grams with high information gain

After pruning, the features were processed to identify those with the most information gain (Breiman et al. 1984, Quinlan 1986 – as cited in Manning & Schultz 1999). The 100 features with the highest information gain were kept.

4.1.4 Correlating Features across Domains

The high information n-grams were then cross-correlated to determine those features that appeared in 2 or more of the 4 domains. This resulted in the selection of the following 26 n-grams: **“a great” , “amazing”, “back”, “bad”, “bit”, “buy”, “called”, “easy”, “ excellent”, “ favorite”, “for the”, “good”, “great”, “highly recommend”, “is a”, “love”, “n t”, “perfect”, “recommend”, “service”, “small”, “the best”, “time”, “wait”, “wonderful”, “worth**”.

4.1.5 Constructing Classifiers for each domain and testing for model saliency

Finally, models were trained using SVM, Logistic Regression and Naïve Bayes. These models were subsequently tested on the datasets from the other domains.

4.1.6 Motivation for this technique

Arriving at this approach involved a combination of ideas from the previous work. Furthermore, it was augmented by a series of tests that entailed doing part-of-speech-tagging, variable length n-grams, less strenuous feature set reduction and careful observation of misclassified experiments.

4.2. Mapping Target Words to Source Words

We developed an experimental technique for domain adaptation that involved mapping words in the target domain to words in the source domain. We then transform the test documents, given our by mapping words that occur in the target documents to their corresponding words in the source documents. Then using a classifier trained on the source domain, we make our predictions on the transformed test documents, using our target sample.

We use the Lin similairty criterion [7] (explained below), added to the KL divergence between two word score distributions to get our measure of similarity.

The Lin similarity criterion utilizes the notion of mutual information between words. Formally, mutual information is a measure of dependence between two random variables, given their probability distributions.

Lin similarity uses the distribution over the types of grammatical relations that words occurred to test mutual information. We view a grammatical relation as a tuple <gov, r, dep> where the gov is the governing word of the relation, the r is the type of grammatical relation, and dep is the dependent word of the relation. Hence a distribution for a given gov ranges over all r, dep. The MAP estimate for the mutual information between words w and w’ is log(||w, r, w’|| ||*, r, *||/(||w, r, *|| ||*, r, w’||). Here the || || brackets do not represent the norm, but rather denote the count of how often the w, r, w’ relation occurred, and * here represents kleene *, or the set of all items in the tuple at the given position.

The mutual information between w and w’ through relation r is denoted I(w, r, w’). T(w) is the set of pairs (r, w’) such that I(w, r w’) is positive.

Lin similarity is defined as

$$\frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)}(I(w_1,r,w)+I(w_2,r,w))}{\sum_{(r,w)\in T(w_1)} I(w_1,r,w)+\sum_{(r,w)\in T(w_2)} I(w_2,r,w)}$$

To obtain the counts of grammatical relations that given words occurred in we syntactically parsed sentences with the Stanford Parser. We used a fine-grained notion of a word for this task. Since words can be polysemous (e.g. light can be a noun or a verb) we defined a word to be a string with its attached POS given by the Stanford parser.

This made our feature space much more sparse, but allowed us to distinguish between different senses of words. Words in the target domain were annotated with a "target" boolean so that their distributions would not affect one another. For example, a word w in one domain might take on a different meaning in another domain, and we wouldn't want the relation distributions headed by bold in both domains to be combined, because semantically they may be different. Accordingly the counts for our tuples <w, r, w'>, the gov word (w), was annotated with a target tag, and the r and w' were left the same.

Additionally we distinguished between differently cased words. Our justification was words in all caps carry more sentiment than words in all lower case. Sentiment may not only be conveyed at the word level, but also in the style of the words.

Additionally for our similarity criterion we added in the KL divergence between smoothed distributions of words. Intuitively we want words that occur in similarly scored documents to be more similar than those that do not.

We mapped words in the target domain to the word in the src domain that had the same POS and the maximum similarity according to our metric for that POS.

Using words/POS produces a much sparser matrix than normal. To address this feature sparsity problem we used Latent Semantic Indexing (LSI) [8] to find a low rank approximation for our document/wordcount matrix, to increase the performance of our classifiers. LSI uses Singular Value Decomposition (SVD) to find a low rank approximation (rank =100) that retains the K=100 largest singular values of the Document/Word matrix that is the best approximation of that rank. Each of the LSI features corresponds to linear combinations of original features, or directions in the original vector space.

Using these reduced features we then feed the train our classifiers and obtain predictions for our LSI'ed/transformed test data.

Parsing through all of the reviews was extremely time consuming. As a result we used 4220 of the non-empty reviews from the TripAdvisorHotel corpus, 622 target training examples from the AmazonElectronics corpus, and 674 examples for testing. These were reviews after parsing for a while, with reviews with no text taken out. To choose the target examples for this, we took 30% of the electronics and designated those as the target electronics, and then performed regression using a slack SVM with a radial kernel, and a regression tree (non-boosted). The justification for taking 30% of the products is that each product represents its own distribution in and of itself, so it was not harmful to give all of a products distributions in training, because in testing it was estimating the remaining y's from these new, different distribution that still shared the same "meta" target distribution.

Overall here is our algorithm.

1. Get the counts ||w, r, w'|| and score distributions p(x) for every word, with governing words annotated with a target/source tag. These counts are gathered from the Target and Source sets.
2. For each target word T, generate a mapping to S', such that S' = argmax(KL(p(T), p(S)) + LinSim(T, S)), where S is a word in the source vocabulary.
3. Transform the Test set of documents using the mapping each word T to its corresponding word S'. If we have not seen the word, generate mapping to the empty string.
4. Perform LSI on the document/wordcount matrix of the source to get reduced feature set on 100 features.
5. Map the source and target documents to the new features using LSI.
6. Train a classifier on the transformed source features, test on the transformed test features.

4.3. Other Techniques Explored

In addition to the aforementioned approaches, some baseline assessment was done that simply utilized single word tokens, and single word tokens with part-of-speech tags. Furthermore, we attempted a battery of machine learning techniques including: Bayesian Networks, Decision Trees, Linear Regression, Ensemble Learning and Perceptron.

5. Results

Baseline results for each domain and the mixed dataset are featured below. The test set in each instance was derived from a 95:5 training:test split. The features vectors consisted of lowercase words that appeared at least twice in a given dataset. Finally, the baseline results were computed using multinomial Naïve Bayes.

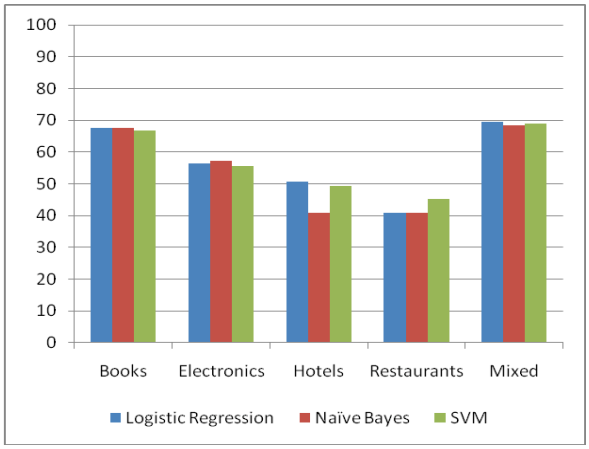
Dataset	Accuracy	Precision	Recall	RMSE
Books	67.3624	0.641	0.673	0.3215
Electronics	78.9969	0.79	0.79	0.2704

Hotels	60.7328	0.601	0.607	0.3518
Restaurants	52.0449	0.512	0.52	0.3613
Mixed	70.8661	0.722	0.709	0.3282

5.1. Generalized N-gram Features

5.1.1 Comparing the Training Accuracy of Different Machine Learning Algorithms on the Generalized Feature Set

The following chart describes classifier accuracy when training data and test data were data were derived from the same dataset. The features in these tests were the 26 generalizable features from section 4.1.4.



5.1.2 Domain Adaptation Efficacy

Using models trained from a given source domain, we performed text classification on the other three domains using Naïve Bayes, SVM and Logistic Regression. The following results are organized based on the target domain and highlights only the algorithm that produced the best results. A full table of these results can be found in the appendix.

5.1.2.1 Books

Book reviews were properly classified at accuracies of 63.9%, 59.22%, and 44.9% when Electronics, Hotels and Restaurants were used as the source domain, respectively. Recall that when Books were used as both the source and test domain, classification accuracy was approximately 67%.

5.1.2.2 Electronics

Electronics reviews were properly classified at accuracies of 63.7%, 59.2%, and 44.9% when Books, Hotels and Restaurants were used as the source domain, respectively. Recall that when Electronics were used as both the source and test domain, classification accuracy was approximately 57.2%.

5.1.2.3 Hotels

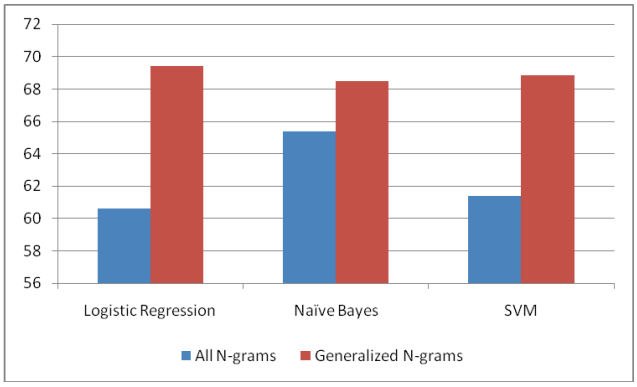
Hotels reviews were properly classified at accuracies of 46.0%, 46.2%, and 45.1% when Books, Electronics and Restaurants were used as the source domain, respectively. Recall that when Hotels were used as both the source and test domain, classification accuracy was approximately 50%.

5.1.2.4 Restaurants

Restaurants reviews were properly classified at accuracies of 35.5%, 35.5%, and 39.15% when Books, Electronics and Hotels were used as the source domain, respectively. Recall that when Restaurants were used as both the source and test domain, classification accuracy was approximately 45.3%.

5.1.3 Classification Accuracy on a Mixed Dataset

Finally a comparison was made between the generalized feature set (the 26 n-grams) and all n-grams for the mixed dataset. Once, again, all n-grams include every n-gram that occurred at least twice in the dataset.



5.2. Target to Source Mapping

Our experimental technique ended up performing poorly with TripAdvisorHotel as the source data, and AmazonElectronics as the target data. Our regression SVM with a radial kernel had an average residual-squared error of 1.667409, with a training residual squared error of 0.4421759, whereas our regression decision tree had an average residual-squared error of 1.893869, with 1.048248 training error.

6. Discussion

6.1. Error Analysis

6.1.1 Variance across Product Domains

As can be immediately deduced from the results, there is a large amount of variance in the different product review categories. This is observed in the differences in accuracy when training and testing within the same domain. The fact that one dataset attains an accuracy of 52% , while another achieves nearly 80% indicates that there is greater variance in restaurant reviews as compared to electronics reviews. One possible reason for this may be related to the lexical diversity of book reviews as compared to restaurant diverse. Though it is somewhat counter intuitive, and may actually reflect the large number of different authors and book titles that appear within book reviews, it's possible that having a large vocabulary that reviewer's use helps them be more consistent and less ambiguous in expressing their sentiment. Accordingly, more exact natural language understanding may be needed to properly differentiate meaning from restaurant and hotel reviews. This problem may be further complicated by the relatively large number of possible items that one can attain at various restaurants as compared to the relative consistency in the types of electronics that people use. Moreover, the reviews for restaurants are consistently longer than most electronics reviews.

An additional consideration in this regard, which can perhaps be considered a finding of this study, is that people seem to have much more consistency around the quality and utility of electronics than they do around gastronomical preferences.

One final consideration may have to do with the nature of the reviews. In the case of hotels we found that it is not uncommon for people to provide a thorough description of the amenities on-site, while also including comments about the surrounding area, and places that one can walk to. Again, this points to a larger scope in the types of things that people will talk about in hotel and restaurant reviews as compared to electronics and book reviews.

6.1.2 Different Performance Across Machine Learning Techniques

Moving now to the results presented in section 5.1.1, the reader can observe that the data confirms our previous supposition that no single machine learning algorithm can consistently maximize performance across domains. The performance is not only feature specific, but is also domain specific. This variance is likely related to both the type of features that we selected and the number of reviews that we observed.

6.1.3 Efficacy of Generalized N-Grams

It comes as little surprise that restricting the feature set to only generalizable n-grams, will reduce the overall classification accuracy within the source domain. This makes sense as many of the features that are central to realizing domain specific class differences are eliminated. What is interesting, however, is the magnitude in the decrease in performance in the different domains. The trend within this data mirrors the trend that we see in the baseline accuracy of each domain. This would suggest that many of the language factors that influenced baseline accuracy (vocabulary size, variability in sentiment consistency, etc.) how well the generalizable features could be applied to that domain.

6.1.4 Efficacy of Domain Adaptation via Generalizable Features

Observing the efficacy of domain adaptation, ie when we have different train and test domains, we again see a large amount of variance. In the case of using Electronics as the source domain and Books as the target domain, we see a relatively comparable classifier as the Books-Books classifier. However, the opposite setup, source – Hotels, Target- Electronics, actually produces a superior classifier for Electronics. To explore why this might be the case, we examine the numerical coefficients for the logistic regression model for each.

Electronics Generalized Classifier

Variable	3	1	4	2
a great	-0.5141	-0.2467	-0.5092	-0.5472
amazing	-1.2624	-3.4481	-0.7056	-18.3434
back	0.9573	1.4443	0.1054	0.9699
bad	0.9137	1.5788	0.6168	1.2392
bit	0.1229	-0.2214	0.3499	-1.1292
buy	0.5097	0.9324	-0.1097	0.0091
called	1.1476	0.9458	0.5857	0.6687
easy	-0.9947	-1.2813	-0.2835	-1.1303
excellent	-1.631	-0.2384	-0.1667	-1.1382
for the	0.3532	-0.0569	0.1678	0.4883
good	0.2099	0.2637	0.7638	-0.0117
great	-0.0426	-1.5045	-0.0464	-0.833
is a	0.4442	0.3352	0.3815	0.3899
love	-1.3448	-2.1406	-0.5031	-1.4471
n t	0.034	0.0584	0.0209	0.7233
perfect	-0.7359	-1.2252	0.0624	-1.0546
recommend	-0.5924	-0.6977	-0.5099	-1.7667
service	-0.2771	1.3804	-0.273	1.1344
small	0.0702	-1.9254	0.0354	-0.469
the best	-0.3945	-0.9758	-0.3561	-1.5546
time	0.0908	0.3905	0.1532	0.2771
wait	0.4765	0.9298	-0.199	0.8631
worth	-0.3105	-1.3558	-0.5457	-0.0408
Intercept	-1.9668	-2.0807	-1.0086	-2.361

Books Generalized Classifier

Variable	3	1	4	2
a great	-0.0526	-1.058	-0.0719	-0.5021
amazing	-1.0577	-2.2504	-0.9653	-1.2979
back	-0.0609	0.049	-0.0357	-0.1082
bad	0.4174	0.8457	0.2991	0.8014
bit	0.6239	-0.5978	0.8739	0.8071
buy	-0.3841	0.5825	-0.3819	0.4233
called	0.2236	0.8798	-0.0053	-0.3931
easy	-0.3552	-1.5976	-0.3098	-1.4032
excellent	-0.8541	-1.4372	-0.2234	-0.833
favorite	-0.4605	-0.8875	-0.2867	-24.3048
for the	0.2841	0.2289	0.1488	0.1705
good	0.8058	0.4409	0.6342	0.6076
great	-0.3099	-0.1988	-0.0166	-0.1813
highly recommend	-2.0929	-1.1718	-0.6499	-16.8274
is a	0.0052	-0.2086	0.243	-0.3073
love	-0.223	-0.3306	-0.246	-0.1681
n t	0.7546	0.821	0.4137	0.8729
perfect	-0.3599	-1.6347	-0.7673	-0.4744
recommend	-0.1187	-0.5198	0.0627	-0.3319
service	-0.3919	-0.5988	0.1099	-0.383
small	0.0074	-0.3798	-0.1891	-0.3586
the best	-0.7156	-0.905	-0.3377	-0.7134
time	0.1891	0.236	0.0546	0.1144
wait	-0.0141	-0.1928	-0.6345	-0.3766
wonderful	-0.9403	-2.0878	-0.7017	-1.5462
worth	0.5403	-0.119	0.6022	0.1435
Intercept	-2.3555	-2.2427	-1.6096	-2.641

Looking at these numbers, the trends are generally the same, with words related to positive sentiment, like "amazing" being more indicative of a 4-star rating than a 1-star rating. However, there are a number of subtleties that appear in the form of non-sequential trending. For example, when one observes "good" in the Electronics classifier, a 2-star rating is considered less "good" than a 1-star rating. The supposition here is that words of positive sentiment are being utilized in a negative fashion more frequently in 1-star ratings than in 2-star ratings. We also saw examples where "good" items and services are described as a way of providing a comparison to reasonable expectations. For example, one electronics 1-star rating contained

"…but I assumed that a good product, with good parts, from a company that desires to satisfy its customers with a good QA/QC program to ensure that good products were produced, would last longer than a year and a half."

Another example of where positive sentiment is injected into negative reviews that we commonly observed was related to customers wishing to start, or end, a scathing review with a positive note. A couple of examples are:

"I had previous good products from viewsonic but I dont have the same faith in them anymore."

"This monitor was great for the 11 months it worked for me."

Beyond this, we see that Hotels and Restaurants, neither of which represented very tight internally invariant domains, performed poorly as classifiers for Books. This can likely be attributed to the highly variable nature of the content of reviews, as mentioned in section 6.1.1.

The last deviation that we'd like to identify relates to the particularly poor performance that we observe for all testing of the Restaurants data, with a specific focus on why Hotels served as a superior source domain than Books or Electronics. While one can argue that there is not much difference between 39% and 35%, we account for this difference by considering the relative proximity of Hotel and Restaurant reviews. Many factors that people consider are the same, and are described in the same context. People will often comment on the ambiance, service, cleanliness and hospitality. These are ideas that only occasionally come up in electronics and book reviews, and are normally in the context of customer support.

Our experimental technique, while attempting to map features in the target space back to the source space in a reasonable manner failed. This could be for a couple of top level reasons. First, this is akin to doing Machine Translation at the word to word level. It might be that the right level of translating features between the target and the input space should be done at a higher level that looks at combinations of source/target word mappings, rather than an intra-word mapping, and uses the target examples for cross-validation. Additionally moving to LSI features made interpreting the models difficult. Since there were 100 features we selected, each being directions in the original feature space, it made it hard to debug the entire model at once. However, our translation approach did provide for interesting error analysis. At a top level, we can view the original and transformed documents to see what semantic nature was and was not preserved. We analyze the following test AmazonElectronic review of score 1 to get a sense of what our model missed.

1. Within/IN a/DT half/NN hour/NN all/DT of/IN the/DT programs/NNS locked/VBD up/RP ./.

Translation

Each/DT fwy/NN compromise/NN Each/DT Constant/IN Half/DT rodents/NNS

2. Tried/JJ everything/NN ./.
Translation
fwy/NN

3. NO/DT 800/CD number/NN to/TO call/VB ./.
Translation
26.00/CD fwy/NN to/TO fault/VB

4. Control/NNP ,/, alt/NN delete/NN would/MD not/RB 5. work/VB ./.
Translation
Fred/NNP ,/, need/MD anytime/RB chat/VB

5. No/DT Support/NN at/IN all/DT ./.
Translation
Half/DT fwy/NN Constant/IN Each/DT

6. Returning/VBZ and/CC will/MD not/RB buy/VB again/RB ./.
Translation
v./CC guys/MD anytime/RB Room/VB Overall/RB

7. Why/WRB would/MD a/DT computer/NN manufacturer/NN not/RB even/RB have/VB a/DT number/NN to/TO call/VB ?/.
Translation
where/WRB need/MD Each/DT fwy/NN mouse/NN anytime/RB anytime/RB chat/VB Each/DT fwy/NN to/TO fault/VB

8. VB The/DT worst/JJS company/NN ever/RB ./.
Translation
largest/JJS subway/NN correctly/RB

On each line is a sentence with words annotated with a part of speech tag. The reviewer is clearly fed up with their computer, labeling the company "the worst company ever." We see that the sentiment is moderately carried over to the translated review

This translation is both nonsensical and words are missing. One of the hopeful things about this translation is that we see that "to" (7) is mapped to itself from the target domain, and "a" is mapped to "each" (1). However, these are insignificant for prediction.

The richest, sentiment wise, sentences are (7) and (8). In (7) the rhetorical structure of the translation is not captured whatsoever. Since each word is translated independently, the relationship between "Why would" and negative sentiment is not well captured at all, instead being translated into "where need." This could be remedied by performing translating at a higher level, rather than the word level. In (7) the same bigram style feature is missed with "not even" being translated to "anytime anytime."

The culmination of all of these errors led us to predict 3.42567 for this example, when its true value was 1.

One notable thing about the translation is that we see a lot of things translated to fwy, an abbreviation of freeway, which appeared infrequently in our corpus. One thing to do would have been to restrict mappings only to words that occurred above a certain threshold.

This example reveals some of the problems with our aforementioned approach, in addition to the difficulty of the problem. The main problem was that, in designing our features, we fragmented the space way too much.

We viewed words of different case as distinct, and also tagged words with part of speech. This maps our original feature space of words to Words X PartsOfSpeech X Case. A solution to this problem would be to only use words, or Words X PartOfSpeech, and take a case insensitive approach.

Part of the difficulty lies in the problem. With more target validation data, we could have eliminated the feature sparsity problem, and probably generated a better model.

Key Findings of the Generalized Feature Results

6.1.5 Generalizable Features Have Some Relevance Across Domains

Based on our work we see that generalizable features are capable of capturing a sizable portion of sentiment, relative to our baseline metrics. While it is by no means the case that the generalized feature set is sufficient for doing state-of-the-art predictions across a variety of domains, our results suggest that there are a number of features that generalize to multiple domains. Future work could be employed to establish a more complete set of these general features.

6.1.6 Generalizable Features as a way for improving accuracy within mixed domains

Of particular interest are the finding that we observed in section 5.1.3. These results demonstrate that the generalized feature set that we constructed was consistently superior to using the entire set of n-grams. This result was repeated across all three machine learning algorithms, and may have some useful implications for future work. When working with a dataset that contains information from a mix of domains it is not unlikely to encounter a large number of domain specific factors. However, while these domain specific features help for one subset of the mixed domain, it impedes accuracy for the remainder of items in the mixed domain. Accordingly, by only using features that have consistent behavior across all domains, we are able to improve the accuracy of predictions.

7. Future Work

7.1. Include Greater Semantic Knowledge

While our latter model utilized similarity relations based on word context, and some semantic understanding of the text, this could be enhanced by using a repository like WordNet to determine synonyms between words. This may ultimately help identify more general similarities between documents of different domains.

7.2. Explore Paragraph Structure

One particularity of user reviews that is lost in most analyses is the structure of different reviews. 1 and 2 star reviews may often start with a positive note, and then be followed by several sentences of denigrating comments. It's also not uncommon to also find 1 and 2 star reviews that start out nicely, but end with the customer berating the organization. There is also a tendency in some domains for customers to editorialize and describe both the positive and negative aspects of a product. Augmenting existing techniques with knowledge about the structure of reviews could help improve classifier performance.

8. Conclusion

In this paper we have described two approaches for addressing domain adaptation. Neither technique provided a final solution to this challenging problem, but both provided important glimpses into how to make further headway in the construction of text categorization models that work in multiple domains.

References

[1] Blitzer, J. et al. *Domain Adaptation for Sentiment Classification.* ACL 2007.

[2] Blitzer, J. et al. *Domain Adaptation with Structural Correspondence Learning.* EMNLP 2006.

[3] Daume III, H. *Frustratingly Easy Domain Adaptation.* ACL 2007.

[4] Jurafsky, D. and Martin, J. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Second Edition. Prentice Hall.

[5] Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing.* MIT Press.

[6] Tan, S., Cheng, X., Wang, Y., and Xu, H. 2009. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In *Proceedings of the 31th European Conference on IR Research on Advances in information Retrieval* (Toulouse, France, April 06 - 09, 2009). M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, Eds. Lecture Notes In Computer Science, vol. 5478. Springer-Verlag, Berlin, Heidelberg, 337-349.

[7] Lin, D. *Automatic Retreival and Clustering of Similar Words*. ACL 1998.

[8] Deerwester S. et. al. *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, vol 41, 1990.

Domain Adaptation Results for Generalized Features based on Training Domain and Testing Domain

Algorithm	Train	Test	Accuracy (%)	Precision	Recall	F-Score	RMSE
Logistic Regression	Books	Books	67.5373	0.519	0.675	0.557	0.3143
Naïve Bayes	Books	Books	67.5373	0.55	0.675	0.558	0.3164
SVM	Books	Books	66.791	0.446	0.668	0.535	0.3511
Logistic Regression	Electronics	Books	62.1475	0.483	0.621	0.522	0.333
Naïve Bayes	Electronics	Books	58.5994	0.459	0.586	0.496	0.3437
SVM	Electronics	Books	63.8842	0.408	0.639	0.498	0.3591
Logistic Regression	Hotels	Books	55.3501	0.523	0.554	0.522	0.3497
Naïve Bayes	Hotels	Books	56.4706	0.503	0.565	0.521	0.3479
SVM	Hotels	Books	59.2157	0.481	0.592	0.526	0.2721
Logistic Regression	Restaurants	Books	41.5686	0.506	0.416	0.44	0.3705
Naïve Bayes	Restaurants	Books	44.8926	0.489	0.449	0.463	0.3723
SVM	Restaurants	Books	32.1008	0.507	0.321	0.344	0.3847
Logistic Regression	Books	Electronics	63.6975	0.502	0.637	0.512	0.3219
Naïve Bayes	Books	Electronics	57.9125	0.457	0.578	0.444	0.3397
SVM	Books	Electronics	57.9728	0.336	0.58	0.425	0.3612
Logistic Regression	Electronics	Electronics	56.4	0.454	0.564	0.458	0.3328
SVM	Electronics	Electronics	55.6	0.309	0.556	0.397	0.3594
Naïve Bayes	Electronics	Electronics	57.2	0.553	0.572	0.453	0.3348
Logistic Regression	Hotels	Electronics	50.3806	0.5	0.504	0.485	0.3546
Naïve Bayes	Hotels	Electronics	50.0801	0.484	0.501	0.478	0.3536
SVM	Hotels	Electronics	51.7028	0.453	0.517	0.479	0.3649
Logistic Regression	Restaurants	Electronics	41.5665	0.489	0.416	0.431	0.3766
Naïve Bayes	Restaurants	Electronics	38.5016	0.459	0.385	0.405	0.3794
SVM	Restaurants	Electronics	35.3165	0.485	0.353	0.372	0.3867
Logistic Regression	Books	Hotels	45.6392	0.364	0.456	0.312	0.3775
Naïve Bayes	Books	Hotels	45.9578	0.401	0.46	0.316	0.371
SVM	Books	Hotels	44.8825	0.201	0.449	0.278	0.3734
Logistic Regression	Electronics	Hotels	46.1569	0.409	0.462	0.359	0.3671
Naïve Bayes	Electronics	Hotels	40.3823	0.349	0.404	0.31	0.3795
SVM	Electronics	Hotels	44.8825	0.201	0.449	0.278	0.3741
Logistic Regression	Hotels	Hotels	50.5976	0.407	0.506	0.448	0.3572
SVM	Hotels	Hotels	49.4024	0.377	0.494	0.427	0.3673
Naïve Bayes	Hotels	Hotels	40.8907	0.438	0.409	0.36	0.3654
Logistic Regression	Restaurants	Hotels	45.1414	0.428	0.451	0.432	0.3651
Naïve Bayes	Restaurants	Hotels	42.6523	0.415	0.427	0.419	0.3665
SVM	Restaurants	Hotels	43.2497	0.415	0.432	0.411	0.3762

Logistic Regression	Books	Restaurants	35.5191	0.284	0.355	0.194	0.411
Naïve Bayes	Books	Restaurants	35.5394	0.28	0.355	0.193	0.4038
SVM	Books	Restaurants	35.1953	0.124	0.352	0.183	0.3828
Logistic Regression	Electronics	Restaurants	35.5191	0.292	0.355	0.211	0.3944
Naïve Bayes	Electronics	Restaurants	33.2726	0.263	0.33	0.197	0.3986
SVM	Electronics	Restaurants	35.1953	0.124	0.352	0.183	0.3867
Logistic Regression	Hotels	Restaurants	39.1419	0.301	0.391	0.299	0.3754
Naïve Bayes	Hotels	Restaurants	35.8227	0.31	0.358	0.244	0.273
SVM	Hotels	Restaurants	36.4299	0.258	0.364	0.531	0.3862
Logistic Regression	Restaurants	Restaurants	40.8907	0.34	0.409	0.363	0.3637
Naïve Bayes	Restaurants	Restaurants	40.8907	0.438	0.409	0.36	0.3654
SVM	Restaurants	Restaurants	45.3441	0.174	0.453	0.386	0.3767