Customizing Sentiment Classifiers to New Domains: a Case Study

Anthony Aue and **Michael Gamon**
Microsoft Research
1 Microsoft Way
Redmond, WA. USA
anthaue,mgamon@microsoft.com

Abstract

Sentiment classification is a very domain-specific problem; classifiers trained in one domain do not perform well in others. Unfortunately, many domains are lacking in large amounts of labeled data for fully-supervised learning approaches. At the same time, sentiment classifiers need to be customizable to new domains in order to be useful in practice. We attempt to address these difficulties and constraints in this paper, where we survey four different approaches to customizing a sentiment classification system to a new target domain in the absence of large amounts of labeled data. We base our experiments on data from four different domains. After establishing that naïve cross-domain classification results in poor classification accuracy, we compare results obtained by using each of the four approaches and discuss their advantages, disadvantages and performance.

1 Introduction

In recent years there has been an increasing interest in the detection and classification of sentiment or affect in various text genres (Pang *et al.* 02; Pang & Lee 04; Turney 02; Turney & Littman 02; Wiebe *et al.* 01; Bai *et al.* 04; Yu & Hatzivassiloglou 03). This area constitutes a problem that is orthogonal to the usual task of text classification: In traditional text classification the focus is on topic identification, whereas in sentiment classification the focus is on the assessment of the writer's sentiment toward the topic. Ideally, sentiment classification ought to be able to address fairly sophisticated issues - identifying the object of sentiment, detecting mixed and overlapping sentiments in a text, identifying and dealing with sarcasm, etc. In practice, most work to date has been concerned with the less ambitious goal of identifying the overall polarity of sentiment in a document, i.e. whether the writer is expressing positive or negative opinions. This task by itself has proved to be interesting and challenging enough, and is the framework within which the experiments in the present work were conducted.

Detection of sentiment is an important technology for applications in business intelligence, where customer reviews, customer feedback, survey responses, newsgroup postings, etc. are automatically processed in order to extract summary information. At a time when large companies receive many thousands of pieces of feedback on a daily basis, human processing of such text volumes is prohibitively expensive; the only alternative is automatic extraction of relevant information. Ideally one would like to be able to quickly and cheaply customize a system to provide reasonably accurate sentiment classification for a domain.

This is not a simple problem, since sentiment in different domains can be expressed in very different ways (Engström 04). Supervised classification techniques which are typically applied to the sentiment classification problem require large amounts of labeled training data. Acquisition of these labeled data can be time-consuming and expensive. This paper explores various strategies for training classifiers in domains lacking large numbers of labeled training examples. We present four different strategies to customize sentiment classifiers to a new domain (hereafter: target domain) in the absence of large amounts of labeled data in that domain. The four approaches we investigate and compare are:

1. training on a mixture of labeled data from other domains where such data are available
2. training a classifier as above, but limiting the set of features to those observed in the target domain
3. using ensembles of classifiers from domains with available labeled data
4. combining small amounts of labeled data with large amounts of unlabeled data in the target domain

2 Data

We used data from 4 different sources in our experiments.

- Movie review data ("movie"): for this domain we used the movie review data set made public by Pang and Lee (Pang & Lee 04). The data consist of 1000 positive and 1000 negative reviews from movie databases. This has become the *de facto* standard data set for sentiment classification.

- Book review data ("book"): in this domain we collected 1000 positive and 1000 negative book reviews from the web.

- Product Support Services web survey data ("pss"): the data in this domain consist of verbatim user feedback from a web survey. The data contain 2564 examples of positive feedback and 2371 examples of negative feedback, based on an associated rating of "not satisfied" versus "very satisfied".

- Knowledge Base web survey data ("kb"): these data were collected along the same lines as the previous data set. They consist of 6035 examples of "bad" feedback and 6285 examples of "good" feedback.

These four domains differ considerably in their properties. Movie reviews tend to be lengthy and elaborate; book reviews are shorter but still may consist of multiple paragraphs. The two sets of survey data, on the other hand, consist of typically very short pieces of text (often just phrases, not even complete sentences).

3 Experimental Setup

Each document is represented as a feature vector. The feature sets in our experiments consisted of unigrams, bigrams and trigrams. Only features that occurred 3 times or more in any of the domains were included in the feature vectors for that domain. Ngram features are binary, i.e. only absence versus presence of an ngram in a document is indicated, not the frequency of that ngram. This decision is motivated by various results in the literature where binary features outperformed frequency features in similar tasks (Pang & Lee 04; Joachims 98).

In order to reduce vector size, we employed a cutoff on the training set using the log likelihood ratio (LLR) for each feature with respect to the class feature (Dunning 93). In this feature selection method, only the top n LLR-ranked features were included, where n ranges from 1000 to 10,000. In preliminary experiments this approach proved to yield better results than a simple count cutoff.

Results for the initial round of experiments, described in Section 4, are based on 5-fold cross-validation. Where statistical significance is mentioned, the assessment is based on the McNemar test at a 99% significance level. The McNemar test has proved reliable for the comparison of different classifiers in supervised learning experiments (Dietterich 1998).

In all of the experiments except for the experiments in Section 5.4, we used support vector machines (SVMs). SVMs have consistently been shown to perform better than other classification algorithms for text classification in general (Joachims 98; Dumais *et al.* 98), and for sentiment classification in particular (Pang *et al.* 02; Pang & Lee 04). The training algorithm we used is Sequential Minimal Optimization (SMO) (Platt 99). For the experiments in Section 5.4 we used naïve Bayes classifiers because they can be formalized as generative models whose parameters can be tuned using the EM algorithm.

In the experiments in Section 5, which compare the different strategies for customizing classifiers to domains with little labeled training data, the test data sets for each target domain consist of sets of 1800 randomly chosen test vectors. For approaches capable of using small amounts of labeled target domain data, parameter tuning data sets of 50, 100, and 200 vectors were used. No cross validation was performed for these experiments since no training was done on target domain data.

4 Classification Accuracy Within and Across Domains

In a first set of experiments, we establish a baseline for experimentation with more sophisticated techniques and try to get a sense of the extent of domain specificity and generalizability for the four domains we are dealing with. We trained SVM classifiers for each domain, using four different feature sets (all ngrams, unigrams, bigrams, trigrams) and six different LLR cutoffs (no cutoff, top 20k/10k/5k/2k/1k features). We then tested

	movie	book	kb	pss
movie	**90.45**	70.29	57.59	61.36
book	72.08	**79.42**	59.28	66.59
kb	57.1	58.62	**77.34**	81.42
pss	52.16	55.33	70.48	**83.73**

Table 1: Best results of svm classifiers within and across domains

	movie	book	kb	pss
movie	ngrams, top 20k	unigrams, top 10k	unigrams, no cutoff	unigrams, no cutoff
book	unigrams, top 5k	ngrams, top 2k	unigrams, top 1k	unigrams, top 2k
kb	unigrams, top 5k	unigrams, top 1k	ngrams, top 2k	unigrams, top 2k
pss	trigrams, top 1k	trigrams, no cutoff	ngrams, top 2k	ngrams, top2k

Table 2: Feature sets and LLR cutoffs that produced the best results in Table 1

each of the resulting classifiers on each domain, so that each classifier was tested both on its own native domain and on all three foreign domains. Table 1 shows the best results for each classifier/domain combination. Numbers in boldface indicate the best results within domain, i.e. when both training data and test data were drawn from the same domain. The baseline accuracy (most frequent class value) for the domains are: 50% for the movie and book domain, 51% for the KB domain, and 52% for the PSS domain.

As far as we can tell, the result on the movie data set is the best so far reported in the literature. Our best guess as to the reason for this is the combination between a lower frequency count cutoff for the features than reported in most research, and additional feature reduction by LLR. The latter tends to select features that (even at low frequencies) have a good correlation with the target. Table 2 shows for each of these (best) accuracy results, which LLR cutoff and feature set they are based on.

Table 1 and Table 2 illustrate a number of important generalizations regarding domain specificity of sentiment detection:

- Domain differences are substantial to the point where a classifier trained on one domain may be barely able to beat the baseline in another domain

Target domain	Training domains	accuracy
Movies	books, kb, pss	72.89
Books	movie, kb, pss	64.58
Kb	movie, book, pss	63.92
Pss	movie, book, kb	74.88

Table 3: Classification accuracy of a classifier trained on three domains and tested on the forth domain

- Within a domain, a mixture of all ngram features works best, while across domains sometimes unigrams, sometimes trigrams, and sometimes all ngrams work best

- Among the four domains we are investigating, books and movies on the one hand and the web survey sets (kb and pss) on the other hand form two distinct clusters.

- Domains vary widely in terms of general difficulty. In our case, as can be seen by the in-domain results, the movie domain is the easiest in which to achieve high accuracy, and the kb domain is the most difficult.

5 Approaches to Overcome the Domain Specificity Problem

5.1 Training One Classifier on all Available Data

One straightforward approach to the problem of multiple domains is to train a single classifier using equal amounts of training data from each of the domains where labeled data are available. In the remainder of the paper we will refer to this approach as the *all_data* approach. This all-purpose classifier, being trained on multiple domains, will be less domain-specific than a classifier that has only seen data from one domain. Throughout the rest of the paper we will be using this approach as our baseline. Table 3 illustrates the classification accuracy for each of the held-out domains using a classifier trained on data from the other three domains. These results are based on a feature cutoff of the top 5000 features according to LLR. In order to keep a balance between the different data sets we restricted the training data sets to 2000 vectors each.

5.2 Limiting Features to those Observed in the Target Domain

A small modification of the all_data approach is to limit the features used during training to those that appear in the target domain. In other words, the training data are represented in the "feature vocabulary" of the target domain. In the remainder of the paper we will refer to this as the limit approach. Note that the limit approach requires no labeled data in the target domain. The assumption behind this strategy is that the distribution of target domain features given the class label in the outside domains is similar to their distribution in the target domain. To the extent that this assumption holds, it allows the classification algorithm to make the best possible use of the out-of-domain data since it need not take into account features that never appear in the target domain. This assumption certainly does not hold in the general case for arbitrary sets of domains; one can easily imagine cases where a given feature would be correlated with positive sentiment in one domain and negative sentiment in another. For instance, while the word "small" might be correlated with positive reviews on a web site dedicated to compact cameras, it would most likely indicate the opposite in a forum dedicated to reviewing SUVs. Nevertheless, there is some hope that the distribution of features between certain domains will be similar enough to each other that the out-of-domain data might be used to some advantage. Our experiments, summarized in Table 4, show the mixed results one would expect given the discussion above; in the kb domain, limiting the feature space significantly improved classification accuracy over the all_data approach. In the book and pss domains the results were statistically identical, and in the movie domain the results when limiting features were significantly worse. These results bring up the very interesting question of whether one could predict a priori, given a set of labeled data from a number of different domains and a small amount of labeled data from the target domain, which subset of those domains, or even which subset of training examples from all the domains, has a feature distribution most similar to the target domain, an area we intend to explore in future research.

Target Domain	All_data	Target domain features only
movies	72.89	**59.11**
books	64.58	64.19
kb	63.92	**70.98**
pss	74.99	75.26

Table 4: Classification accuracy when using a feature set limited by the target domain. Boldface numbers indicate differences that are statistically significant at 99.9%.

5.3 Ensemble of Classifiers

Different classifiers can be combined in ensembles where each of the individual classifiers contributes to the overall classification or class probability. An overview of ensemble classifiers can be found in (Dietterich 97). The classifiers in an ensemble can differ along various parameters, e.g. learning algorithm, training data, feature sets, etc. There is also a wide choice of methods for combining the scores or votes from the different classifiers in an ensemble: simple majority voting, weighted voting (where the weight can be determined by the accuracy of the classifier, the strength of its class probability prediction, etc). Finally, the scores of the classifiers in an ensemble can be combined into a new training set for meta-learning. A meta-classifier is trained on that set and calibrates the combination of scores from the individual classifiers on a held-out data set. More details can be found in (Todorovski & Dzeroski 03).

Classifier ensembles are a promising solution for the data bottleneck because they offer a way to reuse labeled out-of-domain data for a new target domain. For each of the available domains (and for distinct feature sets), a classifier can be trained and included in the ensemble. Faced with data from the target domain, the decision has to be made how the individual classifiers in the ensemble work best together to make adequate predictions in the target domain. Using a meta-learning technique, only n parameters (where n is the number of classifiers in the ensemble) need to be tuned on data from the target domain. Since tuning the ensemble through a meta-learning approach can be achieved with a small labeled data set from the new domain, this method provides a low-effort adaptation to a new domain. The fol-

Held-out domain	All_data	50 training cases	100 training cases	200 training cases
movies	72.89	71.72	74.22	74.55
books	64.58	**67.81**	**70.79**	70.49
kb	63.92	**68.38**	**71.65**	72.39
pss	74.88	76.85	**80.03**	80.47

Table 5: Classification accuracy of a meta-classifier at various training set sizes

lowing experiments were conducted with an ensemble of nine classifiers: unigram, bigram, and trigram classifiers for each of the three training domains. These nine classifiers were then applied to the parameter tuning sets from the target domain in order to create a new set of vectors with nine continuously valued features, each representing the output for one member of the ensemble. An SVM meta-classifier was trained using this data set. For testing, the scores of the nine classifiers were collected for the target domain data, and the meta-classifier was applied to those scores.

The results are presented in Table 5, at three different meta-classifier training set sizes: 50, 100, and 200. The baseline results from Table 3 are repeated in this table as a point of comparison.

Statistical significance testing revealed that in this experiment only the difference between using 50 and 100 training cases is significant. Compared to the baseline results in the first column, statistical significance is indicated by boldface numbers. To summarize: with 50 labeled training cases from the target domain, the classifier ensemble performs significantly better than the all_data approach on two out of the four domains (books and kb). When the training size is increased to a set of 100, performance significantly increases in three domains (books, kb, pss). Performance on the movie domain does not increase significantly compared to the baseline. Increasing the size of the training set from 100 cases to 200 cases does not significantly improve classification accuracy.

5.4 Using In-domain Unlabeled Data

In this approach, due to (Nigam *et al.* 00), small amounts of labeled target domain data were combined with large amounts of unlabeled data from the same domain in order to learn the model parameters for a generative naïve Bayes classifier using the Expectation Maximization algorithm (hereafter: EM). An initial, "priming" classifier is trained using the labeled data alone. This classifier is then used to estimate a probability distribution over the class values of the unlabeled training set. The probability distribution of class values for the labeled data is given by the labels, and never changes. The algorithm uses the probability distributions over both labeled and unlabeled training examples to re-estimate the model parameters, and then recompute the expected class probability distribution over the unlabeled training examples. This step is repeated until convergence is achieved, i.e. the difference in the probability of the model parameters and the data between each iteration is less than some small constant ϵ. Nigam et al. found that using another model parameter, λ, to weight the expected counts for the unlabeled training data significantly improved classification accuracy. A description of the algorithm, along with all the necessary parameter estimation formulae, can be found in (Nigam *et al.* 00).

Target domain	Base-line	amount of labeled data		
		50	100	200
movies	72.89	**61.67**	**79.56**	**77.44**
books	64.58	62.48	**71.08**	**76.55**
Kb	63.92	**65.84**	**68.1**	**73.86**
Pss	74.88	**81.79**	**80.75**	**82.39**

Table 6: Classification accuracy in the bootstrapping approach

We ran each experiment at several different lambda values (0, 0.01, 0.1, 0.3, 0.5, 0.7. 0.9. 0.95, and 1.0), but found that the differences between the best lambda and the lambda set to 1.0 were insignificant. Hence, in the interests of simplicity, all results in Table 6 are reported with a lambda of 1.0. In each case, we ran the algorithm until the expected log probability of the parameters and the data changed by less than 0.01.[1]

With 50 training examples, the results are somewhat mixed – the EM approach is significantly better than all_data in two domains, worse in one, and too close to call in the fourth. With 100 or 200 training examples, however, the EM

[1]In the EM experiments, we included the test data in our unlabeled training data. This is justifiable since the class labels are never consulted.

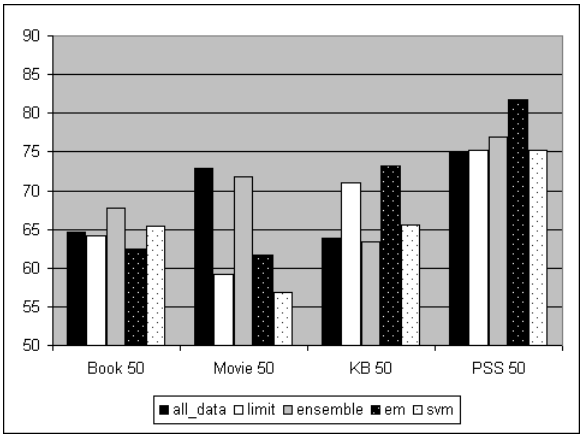

Figure 1: Accuracy with 50 labeled examples

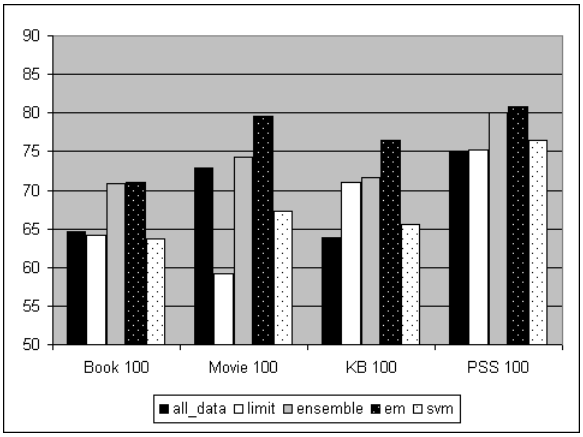

Figure 2: Accuracy with 100 labeled examples

approach is significantly better than the baseline approach in all four domains.

6 Discussion

6.1 Classification Accuracy of Approaches

Figures 1, 2, and 3 compare the classification accuracy of the four classification strategies described in the paper in instances with 50, 100, and 200 labeled training examples. For comparison, the results of applying SVM's trained on similar amounts of data have also been reported. Classification accuracies for the all_data and limit approaches are constant across the charts because these approaches do not use labeled target domain data. While classification accuracy across domains varied somewhat with 50 labeled documents, in both the 100 and 200 labeled document sets the EM approach was best in each case, followed by the ensemble approach. We speculate that the EM approach worked best because it was

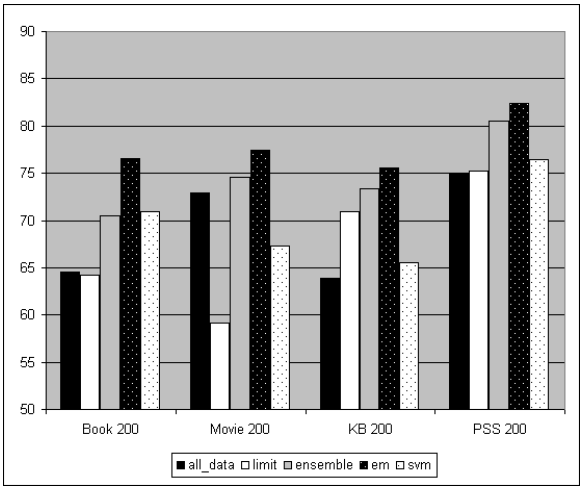

Figure 3: Accuracy with 200 labeled examples

Approach	In-domain, labeled	In-domain, unlabeled	Out-of-domain
Traditional	Lots	None	None
Baseline (train all)	None	None	Lots
Limit	None	None	Lots
Ensemble	Some	None	Lots
NaiveBayes-EM	Some	Lots	None

Table 7: Data Needs

the only approach that was able to make use of the unlabeled data in the target domain. Similarly, the superior performance of the EM and ensemble approaches over the all_data and limit approaches can be attributed to the fact that they are able to take advantage of the labeled data in the target domain, while the other two approaches use only out-of-domain data.

6.2 Data Needs

The different classification approaches in Section 5 have different data requirements. Table 7 summarizes which kinds and amounts of data are needed by each of the different classification strategies.

6.3 Performance

The runtime time and space requirements of the naïve Bayes and SVM classifiers are both roughly linear in the number of features. However, the training time for the SVM classifiers was usually much faster than for the naïve Bayes classifier. Although the naïve Bayes classifier usually converged within 20 or so EM iterations, in some

cases it took more than 100 iterations to reach convergence, which could take up to several hours on data sets with large numbers of features. The SVM classifiers generally converged within a few minutes at worst, and often within seconds.

7 Conclusion

Our survey discussed the challenges inherent in customizing sentiment classifiers to new domains, as well as four possible approaches to the problem. Although all of the approaches differ with regard to what kinds of data they can use, they all share the property that they need, at most, a relatively small number of labeled training examples. The EM approach, since it is able to take advantage of unlabeled data in the target domain, provided the best classification accuracy of the four.

References

(Bai *et al.* 04) Xue Bai, Rema Padman, and Edoardo Airoldi. Sentiment extraction from unstructured text using tabu search enhanced markov blanket. In *Proceedings of the International Workshop on Mining for and from the Semantic Web*, pages 24–35, 2004.

(Dietterich 97) Thomas G. Dietterich. Macine learning research: Four current directions. *AI Magazine*, 18:97–136, 1997.

(Dumais *et al.* 98) Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM*, pages 148–155, 1998.

(Dunning 93) Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74, 1993.

(Engström 04) Charlotte Engström. Topic dependence in sentiment classification. Unpublished M.Sc. thesis, University of Cambridge, 2004.

(Joachims 98) Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML 1998*, pages 137–142. ECML, 1998.

(Nigam *et al.* 00) Kamal Nigam, Andrew McCallum, and Sebastian Thrun. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3):103–134, 2000.

(Pang & Lee 04) Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL 2004*, pages 217–278. ACL, 2004.

(Pang *et al.* 02) Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP 2002*, pages 79–86. EMNLP, 2002.

(Platt 99) John Platt. Fast training of svm's using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernal Methods: Support Vector Machine Learning*, pages 185–208. MIT Press, 1999.

(Todorovski & Dzeroski 03) Ljupco Todorovski and Saso Dzeroski. Combining classifiers with meta decision trees. *Machine Learning*, 50:223–249, 2003.

(Turney & Littman 02) Peter D. Turney and M. L. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Research Council of Canada, 2002.

(Turney 02) Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. pages 417–424, 2002.

(Wiebe *et al.* 01) Janyce Wiebe, Theresa Wilson, and Matthew Bell. Identifying collocations for recognizing opinions. In *Proceedings of the ACL/EACL Workshop on Collocation*. ACL, 2001.

(Yu & Hatzivassiloglou 03) Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP 2003*. EMNLP, 2003.