Domain specific sentiment analysis using cross-domain data

Marcello Hasegawa and Praveen Rokkam

Abstract

Sentiment analysis is a tool used to guide critical business and engineering decisions. Such applications demand sentiment analysis specialized in specific domain areas. In this work we investigate the effectiveness of standard machine learning techniques and feature engineering that can effectively be applied in the industry. Further we investigate the use of labeled data from multiple domain areas to improve the results in a specific domain area in a product review setting. Finally we propose a method based on dimensionality reduction and clustering to reduce the size of the problem without loss in performance.

1. INTRODUCTION

Sentiment analysis is a well-established tool in the industry. Besides consumer applications, sentiment analysis has place among various applications in companies which have been using it as a tool to guide critical business and engineering decisions [1]. Such applications demand sentiment analysis specialized in specific domain areas. In this work we explore domain specific sentiment analysis.

The goal of sentiment analysis is to quantify positivity or negativity from subjective text data. Positivity or negativity can be expressed as a range of discrete values or polarity. Polarity typically is expressed as positive, neutral or negative. Further, sentiment analysis can be scoped around documents, sentences or target particular aspects in text. In this work we address the problem of detecting sentiment in product reviews. This particular application restricts the problem to short text cases. In addition we consider only positive and negative sentiment removing possible ambiguities from neutral reviews. Typically sentiment analysis is approached through the use of sentiment lexicons or as an NLP and statistical classification problem [2]. We rely on the later approach.

While the state of the art research have been pushing the performance in sentiment classification, simpler techniques yielding good results are of great value for industry applications where quality, specific scenarios and ease of implementation are constraints. In this work we consider four different domain areas in product reviews.

A common assumption in machine learning is that the training data is drawn from the same distribution as the unseen test data. In this work, we will explore the assumption that the features we employ for product reviews sentiment classification, in different domains, come from similar distributions. Within this context, we explore two ways of leveraging out-of-domain data for model training. Particularly we propose a method to reduce the amount of necessary training data when augmenting the training set with out-of-domain data by the use dimensionality reduction and clustering.

2. DATA DESCRIPTION

We use the Multi-Domain Sentiment Dataset (version 2.0) [3]. This dataset is a five star rating product review dataset from Amazon.com. The dataset contains multiple domains where each domain is a product area. In this work we considered four domains (Apparel, Music, Video and Electronics). Each domain was separated into positive and negative sentiment reviews. Ratings one and two were considered negative labels and ratings four and five were considered positive labels. We disregarded reviews with rating three. Column two of Table 1 shows the dataset sizes for each domain. Both positive and negative labels are approximately balanced within the domains.

3. MODEL DESCRIPTION

3.1. Learning algorithm

Linear models are known to perform well with bag-of-words features on text. We considered support vector machine (SVM) with a linear kernel and regularized logistic regression models [6]. We chose the regularized logistic regression model as it has lower computational cost for large number of features and had a better performance on prediction accuracy and F-scores. Optimal

Table 1. Training set sizes

Domain	Baseline	% increase: Cross-Domain	% increase: Cluster-Selection	Cluster-Selection reduction gain
Electronics	1,998	300%	176%	-124%
Apparel	2,000	299%	173%	-127%
Music	1,995	299%	110%	-189%
Video	1,995	301%	161%	-140%

parameters for the models were chosen via parameter sweep and ten-fold crossvalidation. For logistic regression we considered a range of regularization coefficients and both L1 and L2 norms. L2 regularized logistic regression performed better.

3.2. Features

The traditional bag-of-words feature modeling approach typically employs n-grams and skip-grams. A number of publications reported good results using bigrams for sentiment classification. We obtained better results using both:

- One-grams, bi-grams and tri-grams combined for review title and text body.
- One-grams of POS tags for text body.

All features were treated to limit sparsity problems. These features result in high dimension feature spaces proportional to the size of the vocabulary. Column two of Table 2 shows the feature set sizes for different domains.

4. DATA SELECTION

4.1. In-Domain Baseline

We start with baseline models trained solely on the datasets corresponding to each individual target domain.

4.2. Cross-Domain

In order to achieve better performance we add out-of-domain data to the target domain training set. The approach consist in splitting the data for one particular domain into training and test sets. To the training set we add the entire datasets from the other domains. The model is evaluated on the test set extracted from the target domain only. This approach resulted in satisfactory improvement to the classification performance across most domains but with the cost of significantly increasing number of features and data volume resulting in a high computational cost. In order to reduce this problem we focused in adding data to the training set selectively while still aiming good performance by exploring similarities on the data.

4.3. Cluster-Selection

A previous work proposes a method to measure similarities among domains [4]. The work describes the use of labeled data from similar domains to adapt classifiers from one domain to another. We propose the use of similar data extracted from several domains simultaneously to enrich the target domain training set in a selectively way. To identify similar data we project one-gram features generated across all domains into a lower dimensional space by use of Latent Semantic Analysis (LSA) [5]. We retain a smaller number of dimensions m and the resulting vectors in this space are clustered using k-means clustering [6]. The input data is the target domain data plus the out-of-domain data. Once a cluster has been assigned to each data observation, the method compute the cluster assignment frequencies within the target domain only. The clusters are ordered in decreasing order of frequency, and the first n clusters are selected. The out-of-domain observations that were assigned to these same clusters are added to the target domain training data. The parameters to be adjusted are the total number of clusters k, the number of LSA dimensions m and the number of clusters to be retained n. The Algorithm 1 illustrates the method. In Algorithm 1, I is the target domain data and O is the out-of-domain data. The procedure FrequentClusters receives the cluster assignments corresponding to the target domain data I and return the most frequent clusters in decreasing order of frequency. The procedure SelectData receives the out-of-domain data O and select all observations whose cluster assignments were the top n most frequent clusters found in the target domain data.

Table 2. Feature set sizes

Domain	Baseline	% increase: Cross-Domain	% increase: Cluster-Selection	Cluster-Selection reduction gain
Electronics	42,073	286%	90%	-196%
Apparel	25,631	551%	191%	-361%
Music	48,923	230%	20%	-210%
Video	68,494	128%	22%	-107%

Algorithm 1 Cluster-Selection

```
procedure SELECTDATA($I$,$O$,k,m,n)
    $Features := \text{GenerateFeatures}(I+O)$
    $Projected := \text{LSA}(Features)$
    $Clustered := \text{KMeans}(Projected[1:m],k)$
    $\{c_1,...,c_k\} := \text{FrequentClusters}(Clustered[I])$
    $Selected := \text{SelectData}(O,\{c_1,...,c_n\})$
    Return $I+Selected$
end procedure
```

5. RESULTS

Table 3 shows the accuracy and F-scores for models trained on the baseline in-domain datasets, the cross-domain enriched training datasets using all data available and the data selected training sets using the cluster-selection algorithm described in the previous section.

5.1. Sentiment classification

Each domain was individually tuned for performance using simple crossvalidation. The apparel domain resulted in the best accuracy, 0.905, whereas the music domain displayed the lowest accuracy, 0.807.

Combining all the domains to the target domain boosted performance. For electronics, accuracy increased 6.5%. Increases in performance were evident in apparel and the video domains as well. For music, we observed a decrease in performance with a drop of 1.4% in accuracy. The average increase across all domains was 2.3% showing that the method can help improve the results.

5.2. Problem size reduction

Despite the gain in performance, adding out-of-domain data to the training set results in larger number of features as the vocabulary increases. This comes with a cost in feature generation time and model training time. To illustrate this, when adding out-of-domain data, the electronics domain sees an increase of 300% in training set size (Table 1). Across all domains, the average increase in training set size was 300%. Regarding number of features, the electronics domain had an increase of 286% (Table 2). The average increase in feature set size across all domains was 299%. A detailed view is presented in Tables 1 and 2.

The cluster-selection algorithm reduced training and feature set sizes considerably. For instance, in the electronics domain we saw decreases of 124% in the training set size and 196% in the feature set size when compared to cross-domain numbers. When using cluster-selection, the average decrease in training set size was 145% and the average decrease in feature set size was 219%.

6. DISCUSSION

The cluster-selection approach described in 4.3 was able to reduce the number of features while keeping comparable performance to the cross-domain approach where we simply add all data available from other domains. The cluster-selection improved the performance with respect to the baseline for three of the domains. The exception was apparel. The accuracy obtained for this particular domain was 10% higher in average than the accuracy for other domains and it seems this result is nearing the limits imposed by the model and features we are considering.

The advantages of cluster-selection is the reduction of the number of features. Its cost overhead for the domains considered was limited to a maximum of 20,000 features to be projected into a lower dimensional space through LSA. The computational overhead, besides the SVD decomposition cost, includes 10 starts of the k-means algorithm. For the current problem we found this cost to be negligible compared to the cost of running the full model. We believe a problem with a much larger dataset and larger number of features could benefit from this approach were the advantage of the reduction of memory resources outweighs the computational cost overhead.

While working in obtaining the best baseline num-

Table 3. Accuracy and F-Scores

Domain	Training Data	Accuracy	F-Score (+)	F-Score (-)
Electronics	In-domain	0.840	0.830	0.850
	Cross-domain	0.895	0.900	0.890
	Cluster-Selection	0.890	0.890	0.910
Apparel	In-domain	0.905	0.900	0.910
	Cross-domain	0.913	0.910	0.910
	Cluster-Selection	0.893	0.900	0.890
Music	In-domain	0.807	0.810	0.810
	Cross-domain	0.795	0.800	0.790
	Cluster-Selection	0.813	0.800	0.820
Video	In-domain	0.820	0.820	0.820
	Cross-domain	0.848	0.850	0.840
	Cluster-Selection	0.848	0.860	0.830

bers we explored a number of approaches using n-grams. Particularly we found that not all approaches are best suited to the different domains. For instance, while fine tuning the feature selection for the domain electronics, we found that one-grams could hurt the performance of the model in the presence of the words ”hard drive”. While in other domains ”hard” is negative, for this case it is neutral.

In this work we also investigated a few other approaches to extend the available features. Particularly we found that adjectives and adverbs obtained from POS tagging doesn’t add much value when compared to the n-gram approach we adopted. Although with little gain, we concluded that the one-gram POS tagging feature counts we employed helps in certain cases to little computational cost. Two promising areas in which we did little investigation was the use of topic features through Latent Dirichlet Allocation (LDA) [7] and features enrichment with SentiWordNet [8]. LDA didn’t performed better than n-grams for this particular problem. We believe LDA can be particularly useful in longer text sentiment classification problems. SentiWordNet features had a neutral effect on the models and we believe a better tuning could have yielded good results but we decided to scope out this approach given the good performance with simpler features.

7. CONCLUSIONS

We have obtained good results in classifying sentiment from product reviews using standard machine learning techniques. Further we explored the use of aditional out-of-domain data to improve the initial results. Adding data from diverse domains increased the training set size and the vocabulary size resulting in a much large number of features. We devised an algorithm which reduced the number of features in average by 219% while maintaing the performance of the brute-force approach of employing all out-of-domain data.

References

[1] Ronen F. Techniques and Applications for Sentiment Analysis, Communications of the ACM, Vol. 56 No. 4, Pages 82-89.

[2] Pang, B., Lee, L. Foundations and Trends in Information Retrieval 2(1-2), pp. 1135, 2008.

[3] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

[4] Blitzer, J., Dredze, M., Pereira, F. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. Association of Computational Linguistics (ACL), 2007.

[5] Landauer, T. K., Dumais, S. T. A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104, 211240 1997.

[6] Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009.

[7] Blei, David M., Ng, Andrew Y., Jordan, Michael I. Lafferty, John, ed. ”Latent Dirichlet allocation”. Journal of Machine Learning Research 3 (45): pp. 9931022 2003.

[8] http://sentiwordnet.isti.cnr.it/