Domain Adaptation for Sentiment Analysis

Chris Ward
cmward@brandeis.edu

Abstract

This work explores different methods of domain adaptation for sentiment analysis using reviews from Yelp as the source domain dataset and tweets as the target domain dataset. I report the results of mixing training data, using the Easy Adapt method (Daumé III, 2009), and a novel instance weighting method, which is able to outperform Easy Adapt. Since I only use unigrams as features for the sentiment classifier, in some sense, the choice of training data (domain and size) can be thought of as the features used for the sentiment analysis task. The code for the experiments is available at `github.com/cmward/sentiment-domain-adaptation`.

1 Introduction

Domain adaptation seeks to train a model using data from one domain, called the *source domain* and use the model to classify samples from another domain, called the *target domain*. Broadly, there are two situations in which domain adaptation is necessary. The first is when the target distribution $P_t(Y|X)$ differs from the source distribution $P_s(Y|X)$. In this case, the same sample x would receive different labels in the source and target domains. This scenario is related to the task of transfer learning. The second scenario is when the target distribution $P_t(X)$ differs from the source distribution $P_s(X)$. In this case, some sample x would be labelled identically in both domains, but the distribution of the data varies across the domains. The latter of these two scenarios is what I deal with in this report, as it is more generally applicable to sentiment analysis than the former.

To see why this is true, we can imagine that there is a set of positive unigram and bigram features for sentiment analysis that would apply to almost any domain, and likewise for negative features. Words like "great" and "horrible" will almost always be good indicators of positive and negative sentiment, respectively, regardless of the domain of the dataset in which they're found. So the problem of domain application for sentiment analysis becomes dealing with the different domains' data distributions.

In these experiments, I use Yelp reviews as the source domain and tweets as the target domain. This is challenging for two reasons. Firstly, reviews from Yelp are just that: reviews, while tweets tend to be contribution to conversations or unsolicited declarations of people's general opinions. Secondly, the domains have substantial stylistic differences in their prose. While both are written fairly informally, Yelp reviews are much longer on average, follow grammatical guidelines more strictly, and have a stated purpose.

2 Methodology

I experimented with three different techniques for adapting from Yelp reviews to tweets. First, I review the properties of the two datasets used in the experiments, and then review the techniques. Finally I report the results of each experiment.

2.1 Datasets

The source dataset consists of the first 560,000 one or five star reviews from the Yelp academic dataset[1], labeled as negative and positive instances, respectively. The Sentiment140 dataset[2] is used as the target dataset. It contains 1.6 million tweets, distantly labeled using the emoticons in each tweet.

The Sentiment140 test set of 359 hand-labeled tweets is used as the test set, and 1,000 Yelp samples are withheld as the 'validation' set.

2.2 Techniques

For each of the following techniques, I use a MaxEnt classifier with L2 regularization trained using minibatch Stochastic Gradient Descent. Baseline accuracy is computed using a third-party precompiled sentiment lexicon[3] and scoring each test sample based on the number of positive and negative words present.

2.2.1 Combining Training Data

This is the simplest possible technique for domain adaptation. One simply takes labeled instances from both the source and target domain and combines them into a single training dataset. I set an upper limit for the number of target samples at 100,000, to simulate the scenario in which target samples are not plentiful. Results are reported in Table 2 for various different combinations of number of instances taken from each domain.

2.2.2 Easy Adapt

When combining data from different domains, ideally, you want to be able to use as little labeled data from the target domain as possible. This is helpful for when you have very little data from the target domain. Easy Adapt (Daumé III, 2009) is a method for augmenting features to allow the model to learn when a feature is useful only in the target domain, only in the source domain, or in both.

Given a sample $\mathbf{x} \in \mathbb{R}^F$, we define two functions ϕ^S and ϕ^T as follows, where $\mathbf{0}$ is a vector of F zeros, and apply them to data samples from the source and target domains, respectively:

$$\phi^S(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x}, \mathbf{0} \rangle$$
$$\phi^T(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0}, \mathbf{x} \rangle$$

We see that Easy Adapt triples the model's feature space from $\mathbb{R}^F$ to $\mathbb{R}^{3F}$, creating new feature vectors consisting of general, source, and target features.

Results for combining training data with Easy Adapt are reported in Table 3.

2.2.3 Instance Weighting

Instance weighting refers to assigning a weight to a sample before using it to train a classifier. Sophisticated work on instance weighting for NLP applications has carried out by (Jiang and Zhai, 2007). I propose a much simpler method of instance weighting. Before training the model, a sentiment lexicon is built for the source and target datasets by finding the sets of 3,000 words which occur most frequently in positive and negative samples in each dataset. The source and target lexicons, L_s and L_t, are then filtered such that $L_t \cap L_s = \varnothing$. For each sample in the training set, two sets are computed. The first, T_t is the set of tokens in the sample that occur in the target lexicon, and the second, T_s is the set of tokens in the sample that occur in the source lexicon. The weight for the instance is then computed as

$$w = \frac{1}{\max(1, |T_s| - |T_t|)}$$

The weight for each sample can be thought of as a penalty for how representative of source-specific polarity features it is.

The results of using instance weighting are given in Table 4.

3 Results

Unsurprisingly, the best results from each technique are achieved when the largest target training set is used.

Easy Adapt provides gains in accuracy over single dataset and mixed dataset training sets in every training set size combination except 500k source/100k target. Interestingly, at 500k/100k, both simple dataset mixing and Easy Adapt achieve a test set accuracy of 79%. It seems as though once there is

[1] https://www.yelp.com/dataset_challenge/dataset

[2] http://help.sentiment140.com/for-students/

[3] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

# Source	# Target	Features	Test acc	Val acc
-	-	lex	.49	-
0	20k	uni	.59	-
0	60k	uni	.57	-
0	100k	uni	.58	-
0	500k	uni	.69	-
0	1m	uni	.68	-
0	1.6m	uni	**.78**	-
20k	0	uni	.66	.71
60k	0	uni	.68	.87
100k	0	uni	.73	.95
500k	0	uni	.76	.97

Table 1: Results of Training on only Twitter or Yelp data. # Source is the number of samples taken from Yelp data and # Target is the number of training samples taken from Twitter data. Test accuracy is computed on the Sentiment140 test set. Validation accuracy is computed on a held out set of 1000 yelp reviews. The features are unigrams (*uni*) and scores derived from a third-party sentiment lexicon (*lex*).

# Source	# Target	Features	Test acc	Val acc
0	1.6m	uni	.78	-
500k	0	uni	.76	.97
20k	20k	uni	.70	.70
100k	20k	uni	.71	.81
100k	100k	uni	.73	.82
500k	20k	uni	.75	.94
500k	100k	uni	**.79**	.92

Table 2: Results of combining Twitter and yelp training data. The features are unigrams (*uni*).

# Source	# Target	Features	Test acc	Val acc
0	1.6m	uni	.78	-
500k	0	uni	.76	.97
20k	20k	easy	.71	.80
100k	20k	easy	.73	.92
100k	100k	easy	.76	.84
500k	20k	easy	.77	.95
500k	100k	easy	**.79**	.93

Table 3: Results of Easy Adapt on different training set sizes. The features are Easy Adapt unigrams (*easy*) and unigrams (*uni*).

# Source	# Target	Features	Test acc	Val acc
0	1.6m	uni	.78	-
500k	0	uni	.76	.97
20k	20k	IW	.70	.72
100k	20k	IW	.74	.91
100k	100k	IW	.76	.84
500k	20k	IW	.77	.94
500k	100k	IW	**.80**	.92

Table 4: Results of combining Yelp and Twitter data with instance weighting. The features are unigrams with instance weighting (*IW*) and unigrams (*uni*).

# Source	# Target	Features	Test acc	Val acc
500k	0	uni	.76	.97
0	1.6m	uni	.78	-
500k	100k	uni	.79	.92
500k	100k	easy	.79	.93
500k	100k	IW	**.80**	.92

Table 5: Best results on the test set from each technique.

enough training data from the target source, Easy Adapt becomes less efficient and simple data mixing suffices. Easy Adapt and dataset mixing are able to gain a point of accuracy over training on all 1.6 million target training samples using the source data and only $1/16^{\text{th}}$ of the full target training set.

At 500k/100k, instance weighting outperforms both data mixing and Easy Adapt with a test accuracy of 80%. Overall, it achieves very similar results to Easy Adapt. It seems as though instance weighting needs more target training data to be effective, and doesn't max out in performance in the way that Easy Adapt does.

4 Conclusion

These experiments have shown that Easy Adapt and simple instance weighting are effective methods for domain adaptation for sentiment analysis when there is a lack of target training data.

A direction for further research would be refining my instance weighting method. Firstly, it would be informative to experiment with different, and perhaps more sophisticated, methods of creating the target and source lexicons, although a simple experiment would be to change the initial size of the polarity lexicons. Secondly, along the lines of (Jiang and Zhai, 2007), if the method for computing instance weights could be parameterized and learned, the results would probably improve, although it would de-

tract from the method's simplicity, which is arguably its strong suit. The instance weighting method could also probably be extended fairly simply to handle the case where the target training data is unlabeled.

References

[Daumé III2009] Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

[Go et al.2009] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12.

[Jiang and Zhai2007] Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *ACL*, volume 7, pages 264–271.