# Active Transfer Learning Across Domains in Sentiment Analysis

**Arunima Kayath**
UC Berkeley
Masters in Data Science
arunima@berkeley.edu

**Anamika Sinha**
UC Berkeley
Masters in Data Science
anamika@berkeley.edu

## Abstract

We describe an approach for active transfer learning from a source to a target domain by adding selected labeled data from the target domain to the source domain to take transfer loss in domain adaptation to zero. Applying active learning enables us to minimize the amount of data from the target domain that needs to be labeled and added to the source domain. While we illustrate the approach for the task of sentiment polarity prediction, it can easily be applied to many other tasks.

## 1 Introduction

As humans we apply our knowledge in one area to learn things faster in another area e.g it is reasonably easier for a soccer player to learn how to play basketball compared to someone who has not played any sport. Can machine learning models also transfer knowledge from one domain to another?

Creating labeled data is expensive and difficult. So effective utilization of existing labeled datasets can have many practical applications. Transfer learning from a source domain to a target domain enables this.

We will examine the application of transfer learning to the specific task of predicting positive or negative ratings from reviews. In addition, we apply concepts of active learning to make transfer learning more effective.

In this paper, we simulate the scenario where we have a labeled source corpus, an equally large unlabeled target domain corpus, and the option to label select instances of the target and add them to our model. We simulate conditions where labeling is expensive, and there is a need to minimize the cost of labeling. We apply principles of active learning to pick instances to label from the target that are likely to be most effective in improving results when added to our source domain train set.

## 2 Literature Review

Transfer learning or domain adaptation has been studied extensively in Natural Language Processing. Two frequently cited papers are Daume III(2007), and Blitzer et al (2007). In Daume(2007), he creates variables that classify features as general, source specific and domain specific. The approach reduces transfer loss on many tasks vs state of the art at the time. Similarly, Blitzer et al(2006,7) use a mathematical approach to establish structural correspondence between pivot features present in both domains, and their correlations with domain specific features to improve domain adaptation. They reduce the error by 46% over the supervised baseline. Remus(2014) in his PhD thesis, uses multiple similarity measures to measure similarity between domains, and hence transferability. He extends the concept of similarity to pick a sample from the source domain that is similar to the target domain to build a model with less transfer loss. Glorot et al (2011) use neural networks for general feature extraction. They build a Deep Learning system based on Stacked Denoising Auto-Encoders to perform unsupervised feature extraction. The resulting features when used in an SVM to predict sentiment show lower transfer loss than Blitzer(2007)

In all of these cases, while transfer loss is reduced, there is a residual transfer loss. We extend the concept of domain adaptation by applying active learning to add selected labeled data from the target domain to the source domain, to eliminate transfer loss. We pick samples from the target domain that most likely to be effective in improving the model built on the source domain. We referred to the Active Learning Literature Survey, Settles(2010) for active learning techniques.

There's a few that seemed particularly interesting. a) Labeling instances with maximum uncertainty in labeling using source model b) Using instances that are most likely to change the model. c) Using a combination of b and c. We used uncertainty as a metric for picking instances to label and add from the target domain.

In addition, given our large sample sizes, we tested CNNs for active transfer learning. The CNN architecture we used was developed by Kim(2014). It improved over state of the art in multiple sentiment benchmarks.

## 3 Methodology

### 3.1 Data

We used Amazon reviews for four domains: Toys (2.2mn reviews), Video Games (1.3mn), Automobile Products (1.4mn) and Home and Kitchen (4.3mn). The data was obtained from `ttp://jmcauley.ucsd.edu/data/amazon/links`.tml The data was split into train, dev and test data in a 60:20:20 ratio before analysis.

### 3.2 Baseline models

We evaluated three different models - Naive Bayes, SVM and CNN, on the core task of predicting positive or negative rating based on review text. We chose Naive Bayes because it is effective and fast, SVM because it yielded the best results for sentiment polarity rating in several papers, and CNN because it is effective at large samples. We tried all three models with different sample sizes.

For text pre-processing for Naive Bayes and SVM, we used the bag of words assumption and word counts. For CNN, the reviews were converted to word ids, and then 100 dimension Glove embeddings.

Next, we calculated the transfer accuracy for the 3 models from each domain to the other domains. When transferring from source to target, we used two scenarios for pre-processing a. the vocabulary of the source domain to pre-process the reviews of the target domain b. the vocabulary, based on source + target domain. In both cases, we used only source labeled data for the training.

CNN Architecture: The architecture of the CNN we used was developed by Yoon Kim (2014). We converted the reviews to word ids with max length of 150 (80% or more of the reviews were less than 150 words in length), and then to word embeddings using Glove Vectors(100 dim). Note: Kim used word2vec. The convolutional layer had filters of width 3,4 or 5, and height of the embeddings. We had 256 filters of each width vs 100 in the paper. We used more filters as we had full reviews while the paper was dealing with just sentences. For max pooling, we picked the max over time for each filter and filter width. So at the end of the max pooling, we had 256 X 3 features. These were passed through a fully connected layer to make a 2 class prediction. We used cross-entropy as the loss function since it is a classification problem, and it can do finer optimization of the probabilities even when the classification is correct, but the probability of prediction can be improved further. We tested several model hyperparameters but did not see a significant improvement in performance.

To determine model effectiveness, we used accuracy and f1-average. We report f1-average as we had significant class imbalance ( 80% of the reviews were positive). f1-average is the simple average of the f1-score on the positive and negative class.

### 3.3 Relationship between domain similarity and transfer loss

Next, we evaluated whether picking a domain based on similarity could minimize transfer loss.

Transfer loss = in-domain accuracy for a model built on domain A - accuracy for domain A using a model built on domain B. A similar calculation can be done for f1-average.

We estimated domain similarity based on two different metrics, JS Divergence and Cosine Similarity.

JS Divergence is similar to KL Divergence, but better suited to comparing different domains.

KL Divergence =

$$D_{KL}(P \,||\, Q) = \sum_{x} P(x) \log_2 (P(x)/Q(x))$$

JS Divergence =

$$D_{JS}(P \,||\, Q) = 1/2 {*} (D_{KL}(P \,||\, M) + D_{KL}(Q \,||\, M))$$

where M = 1/2(Q+R)

JS Divergence is defined even when Q(x) is 0 for a given word, whereas KL Divergence is not.

For calculating both, we used the word count vectors of each domain using the bag of words assumption. To compare across domains, we first created word count vectors on the vocabulary on the reviews of the 4 domains together.

### 3.4 Techniques for active transfer learning

We estimated how much sample we needed to add to the source domain to get to the same accuracy as a model built on 100% of the target domain sample. We calculated Nts/Nt where

Nts = Number of labeled instances from target domain needed when starting from a model built on a source domain, to get to similar accuracy as a model built entirely on the target domain.

Nt = minimum number of labeled instances from target domain to reach maximum possible accuracy when building a model directly on the target domain only.

For actively picking samples from the target domain to label, we tried two different metrics:

a. Certainty : difference in probability predictions for the +ve and -ve labels for each target domain review using the source domain model. A large difference implies high certainty for classification. Low certainty reviews were the first candidates to label and add from the target domain.

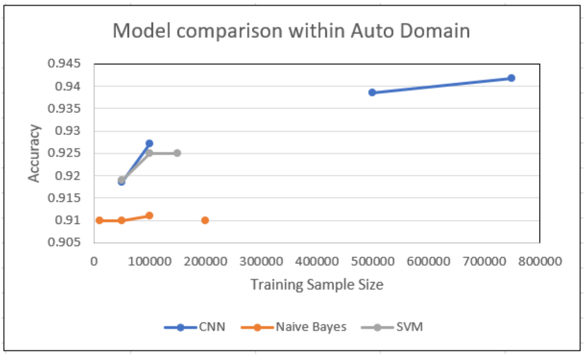
b. Cosine distance at the target instance level : cosine distance of the word count distribution of all reviews of the source domain put together, to the word count distribution of the individual reviews in the target domain.

We could not evaluate active transfer learning with SVM because it took a long time to train with large samples.

## 4 Results

### 4.1 Baseline models, in domain accuracy

The results for in domain accuracy with the three models (Naive Bayes, SVM and CNN) are shown in Figure 1.



As we can see, CNN and SVM did better than Naive Bayes. Naive Bayes showed comparable accuracy at small samples but hit a plateau with increasing sample. CNN continued to improve with large samples. So, we did transfer analysis with large samples with the CNN.

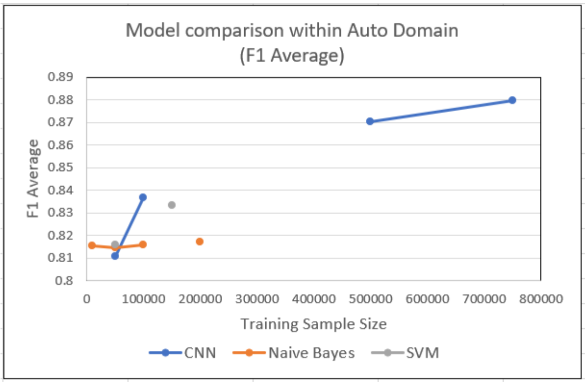


Figure 1. In-Domain model performance comparison with increasing sample size

### 4.2 Baseline models, transfer accuracy

The effectiveness of transfer of a model built on source domain to predict the target domain is shown in figure 2. The scenario shown here is based on the vocabulary of the source domain only. As we can see, SVM does the best, and both CNN and Naive Bayes do not transfer as well.

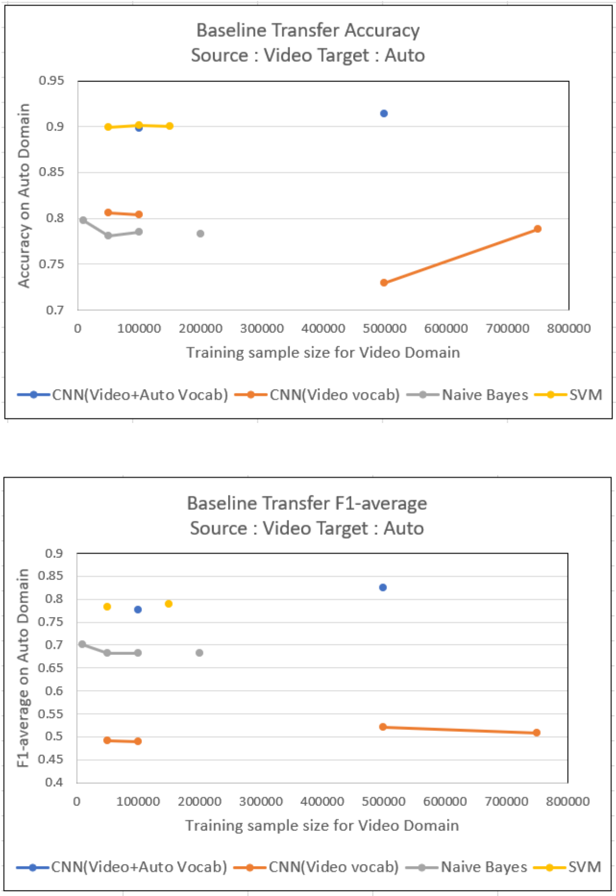


Figure 2. Baseline transfer metrics

The transfer accuracy with CNN improved once we pre-processed the source reviews using the combined vocabulary built on source and target reviews. This uses only the unlabeled sample from the target domain. The CNN was trained on the labeled data from just the source domain. This brought transfer accuracy on par with an SVM. The results are shown in Table 1.

### 4.3 Relationship between transfer accuracy and domain similarity

Figure 3 shows that the transfer loss was approximately correlated with the domain similarity ie transfer loss was lower when transferring from a more similar domain. The relationship is clearer with cosine similarity and for SVM and CNN. Naive Bayes does not show as clear a correlation of transfer loss with similarity. Hence one strategy to minimize transfer loss is to pick a source domain that is most similar to the target domain.

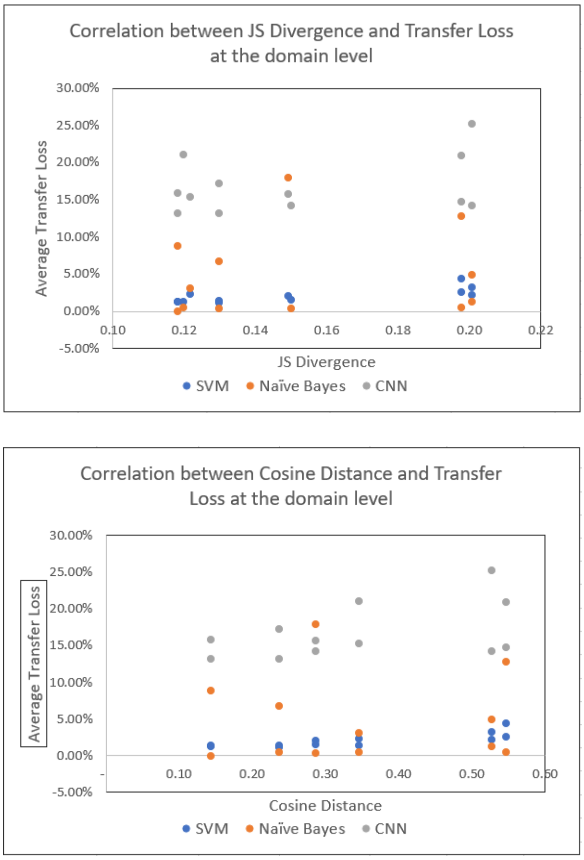


Figure 3. Transfer loss vs similarity metrics

However, there is transfer loss. Next, we explore ways to get to in-domain model accuracy for the target domain, by adding limited labeled data from the target domain to the source domain.

### 4.4 Active transfer learning with Naive Bayes

The analysis with the Naive Bayes model is shown with Video as the source domain and Auto as the target domain as this baseline transfer showed the highest transfer loss.

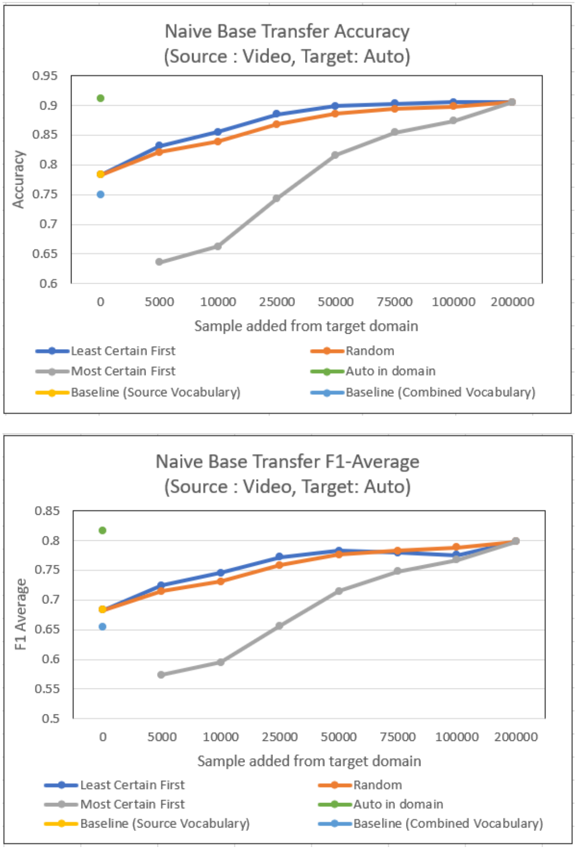


Figure 4. Active transfer with Naive Bayes

As shown in Figure 4, adding sample from the target domain to the source domain improved predictions on the target domain. A 25% sample from the target domain was sufficient to reach peak accuracy with active transfer based on adding the lowest certainty samples first. Peak accuracy was lower than the in-domain model. To confirm the effect, we did the opposite and added target domain samples with high certainty first, and they did not help in reducing transfer loss.

We also tried adding highest cosine distance samples from the target domain first. The results were counter-intuitive at first. We expected target samples with highest cosine distance from the source domain to be most effective. But we found the opposite ie adding target samples with low cosine distance was more effective in reducing transfer loss.

On investigation, we realized this was because we calculated the cosine distance as the distance between the entire source domain and the individual reviews in the target domain. When taking the dot product, most of the values in the source domain get zeroed out, except where the word is present in the target review. So the highest cosine similarity and hence lowest cosine distance were also the longest reviews. Longer reviews are more effective, as they have more information per review. A plot of average review length by cosine distance buckets confirms the hypothesis.

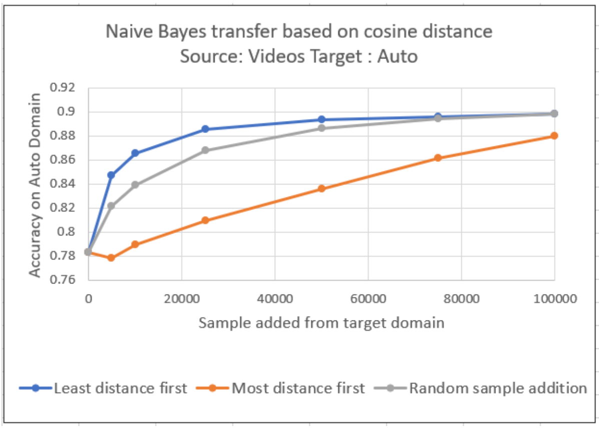


Figure 5. Active transfer based on cosine distance

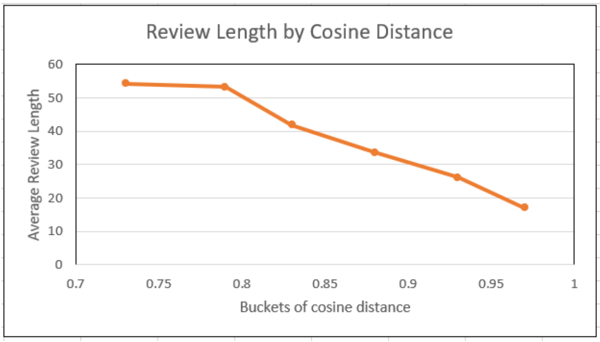


Figure 6. Review length by cosine distance

A better approach to calculate cosine distance is to the take the max of the cosine distance of the target reviews over the individual source reviews. This could be tried in the future.

### 4.5 Active transfer learning with CNN

The results of certainty based active transfer learning with a CNN are shown in figure 6. Active transfer based on adding low certainty samples first outperforms adding random samples. We reach in-domain target model accuracy with just 30% of the target model sample added to the source domain. While not shown, we saw similar results on transfer from Home & Kitchen to Auto.

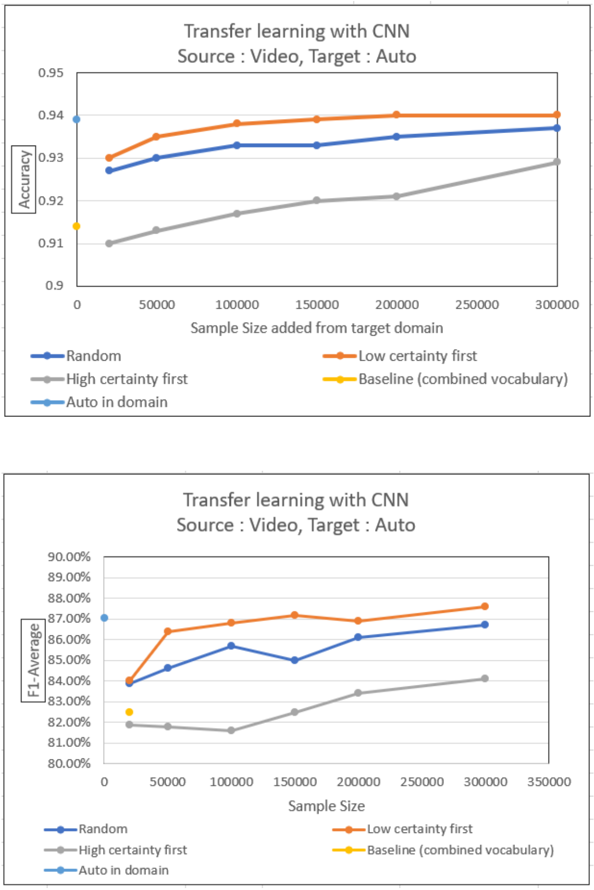


Figure 7. Active transfer with a CNN

Interestingly, just creating a source domain model on the combined vocabulary of the source + target reviews leads to a big improvement in predictions on the target domain, even though the labels are only from the source domain. Shown in table 1 is the accuracy for predictions on Auto, using a model built on Video, with vocabulary using a) source only, and b) using both source and target reviews.

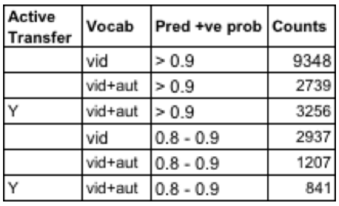
| Source Domain | Source | Source + Target |
|---|---|---|
| Video Games | 72.92% | 91.4% |
| Home and Kitchen | 80.80% | 92.40% |

Table 1: Auto accuracy with source model & vocabulary of a.source reviews b.source+target reviews.

### 4.6 Transfer Error Analysis with CNN

We did error analysis of transfer predictions of the CNN for the model on source vocabulary only compared to the model with combined vocabulary, and the model with addition of 150,000 low certainty samples from the target domain.

We divided the predicted probability into ranges for comparison. The number of instances with high false -ve probability for 3 models (source model with source vocabulary, source model with combined vocabulary, and source + target(150000) are shown below.



| Active Transfer | Vocab | Pred +ve prob | Counts |
|---|---|---|---|
| | vid | > 0.9 | 9348 |
| | vid+aut | > 0.9 | 2739 |
| Y | vid+aut | > 0.9 | 3256 |
| | vid | 0.8 - 0.9 | 2937 |
| | vid+aut | 0.8 - 0.9 | 1207 |
| Y | vid+aut | 0.8 - 0.9 | 841 |

Figure 8. False Positives across three models. Ranges 0.5-0.8 not shown, follow a similar trend.

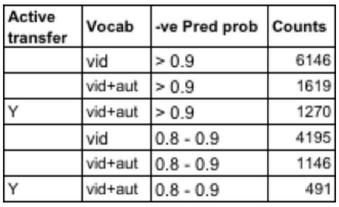
An example of False positives where the models failed.

> “I should have gotten an invoice with the k40 k-30 c b antenna so I can a get a replactment.”

In the above review, the model gets confused by k40 k-30 c b antenna, and -replacement- is misspelled.

> “Amazon’s fitment guide says it will fit a 2006 Jetta. The instructions that come with the blade show the connection system that a Jetta uses. But there is no hole in the wiper blade bracket for the peg on a Jetta wiper arm to slide into.”

In the above review, the model fails when phrases in the sentence are very specific to the domain and have no clear sentiment. Fig 9 below, illustrates number of false negatives across the three models.



| Active transfer | Vocab | -ve Pred prob | Counts |
|---|---|---|---|
| | vid | > 0.9 | 6146 |
| | vid+aut | > 0.9 | 1619 |
| Y | vid+aut | > 0.9 | 1270 |
| | vid | 0.8 - 0.9 | 4195 |
| | vid+aut | 0.8 - 0.9 | 1146 |
| Y | vid+aut | 0.8 - 0.9 | 491 |

Figure 9. False Negatives across three models. Ranges 0.5-0.8 not shown, follow a similar trend.

An interesting false negative sample review.

> “WeatherTech! What more can you say? I had the run of the mill Bug Deflector. I got a deal and when I got it,I got what I paid for,can I say it ”cheap crap” WeatherTech is quality.Not only can you see, you can feel it. I intend to buy more WeatherTech products soon.I know what i’m getting ...”

This is a labeling error by the review author, our active transfer learning model classifies correctly.

With the True Negatives, the combined domain model had a significantly greater proportion of reviews in the higher prediction probability range. We saw a similar trend with the true positives across the three models.

## 5 Conclusion

We show that we can achieve prediction levels as good as an in-domain model for a target domain, using a different source domain supplemented with 25-30% labeled data from the target domain. We selectively label and add data from the target domain where the source model is least certain. We demonstrate that this works with both the Naive Bayes model with small samples (source 100-200K), and the CNN with large source samples (source 0.5 mn).

While we demonstrate the result for sentiment polarity prediction, the approach can be applied to many other NLP or classification tasks with similar limited labeled data in the target domain.

We also found that the CNN was more effective for sentiment polarity classification with the use of unlabeled data from the target domain to create the vocabulary for pre-processing. The CNN seems to abstract to more generalizable features in such cases.

## 6 Potential for further analysis

There are several ways to extend this work further. We can:

- Understand better how the CNN generalizes features when exposed to unlabeled data from the target domain. This was a particularly interesting and surprising result in our work, and we would like to understand this better.
- Explore additional metrics (such as cosine similarity at review level, length of review, reviews with most words out of source vocabulary) to further reduce the amount of target labeled data needed.
- Continued active learning where we add a small labeled batch from the source domain, then repeat the uncertainty prediction with the new model, then add the next batch based on this. This may further reduce the amount of labeled target data needed.

## References

Hal Daume III 2007. Frustratingly Easy Domain Adaptation.

John Blitzer, Mark Dredze, Fernando Pereira 2007 Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification.

Robert Remus 2015. Genre and Domain Dependencies in Sentiment Analysis, PhD Thesis.

Xavier Glorot, Antoine Bordes, Yoshua Bengio. 2011. Domain Adaptation for Large Scale Sentiment Classification: A Deep Learning Approach

John Blitzer, Ryan McDonald, Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning.

Burr Settles 2010. Active Learning Literature Survey.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification.

Ruining He, Julian McAuley 2015. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering.

Julian McAuley, Qinfeng shi, Christopher Targett, Anton van den Hengel 2015. Image-based Recommendations on Styles and Substitutes.

Non-paper References

WildML Understanding Convolutional Neural Networks for NLP. http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/.

WildML Implementing a CNN for Text Classification in TensorFlow. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/.