A Comparative Study of Methods for Transductive Transfer Learning

Andrew Arnold, Ramesh Nallapati and William W. Cohen
Machine Learning Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{aarnold, nmramesh, wcohen}@cs.cmu.edu

Abstract

The problem of transfer learning, where information gained in one learning task is used to improve performance in another related task, is an important new area of research. While previous work has studied the supervised version of this problem, we study the more challenging case of unsupervised transductive transfer learning, where no labeled data from the target domain are available at training.

We describe some current state-of-the-art inductive and transductive approaches and then adapt these models to the problem of transfer learning for protein name extraction. In the process, we introduce a novel maximum entropy based technique, Iterative Feature Transformation (IFT), and show that it achieves comparable performance with state-of-the-art transductive SVMs. We also show how simple relaxations, such as providing additional information like the proportion of positive examples in the test data, can significantly improve the performance of some of the transductive transfer learners.

1 Introduction

Consider the task of *named entity recognition* (NER). Specifically, you are given a corpus of encyclopedia articles in which all the personal name mentions have been labeled. The standard supervised machine learning problem is to learn a classifier over this training data that will successfully label unseen test data drawn from the same distribution as the training data, where "same distribution" could mean anything from having the train and test articles written by the same author to having them written in the same language. Having successfully trained a named entity classifier on this encyclopedia data, now consider the problem of learning to classify tokens as names in instant messenger data. Clearly the problems of identifying names in encyclopedia articles and instant messages are closely related, and learning to do well on one should help your performance on the other. At the same time, however, there are serious differences between the two problems that need to be addressed. For instance, capitalization, which will certainly be a useful feature in the encyclopedia problem, may prove less informative in the instant messenger data since the rules of capitalization are followed less strictly in that domain. Thus there seems to be some need for altering the classifier learned on the first problem (called the *source domain*) to fit the specifics of the second problem (called the *target domain*). This is the problem of *domain adaptation* and is considered a type of *transfer learning*.

The intuitive solution seems to be to simply train on the target domain data. Since this training data would be drawn from the same distribution as the data you will ultimately test over, this approach avoids the transfer issue entirely. The problem with this idea is that often large amounts of labeled data are not available in the target domain. While it has been shown that even small amounts of labeled target data can greatly improve transfer results [4, 6], there has been relatively little work, however, on the case when there is no labeled target data available, that is, totally unsupervised domain adaptation. In this scenario, one way to adapt a model trained on the source domain is to make the unlabeled *target test data* available to the model during training time. Leveraging (unlabeled) test data during training time is called *transductive learning* and is a well studied problem in the scenario when the training data and test data come from the same domain. However, transduction is not well-studied in a transfer setting, where the training and test data come from different domains. Studying transfer learning in a transductive setting will be the main focus of our work.

2 Learning paradigms and related work

Given an example x and a class label y, the standard statistical classification task is to assign a probability, $p(y|x)$, to x of belonging to class y. In the binary classification case the labels are $Y \in \{0, 1\}$. In the case we examine, each example x_i is represented as a vector of binary features $(f_1(x_i), \cdots, f_F(x_i))$ where F is the number of features. The data consists of two disjoint subsets: the training set $(X_{train}, Y_{train}) = \{(x_1, y_1) \cdots, (x_N, y_N)\}$, avail-

able to the model for its training and the test set $X_{test} = (x_1, \cdots, x_M)$, upon which we want to use our trained classifier to make predictions. We discuss below a small subset of the many possible different paradigms of learning associated with the classification problem.

In the paradigm of *inductive learning*, (X_{train}, Y_{train}) are known, while both X_{test} and Y_{test} are completely hidden during training time. In the case of *semi-supervised* inductive learning [20, 16, 9], the learner is also provided with auxiliary unlabeled data $X_{auxiliary}$, that is not part of the test set. It has been noted that such auxiliary data typically helps boost the performance of the classifier significantly.

Another setting that is closely related to semi-supervised learning is *transductive learning* [18, 11, 13], in which X_{test} (but, importantly, not Y_{test}), is known at training time. That is, the learning algorithm knows exactly which examples it will be evaluated on after training. This can be a great asset to the algorithm, allowing it to shape its decision function to match and exploit the properties seen in X_{test}. One can think of transductive learning as a special case of semi-supervised learning in which $X_{auxiliary} = X_{test}$.

In the three cases discussed above, X_{test} and X_{train} are both assumed to have been drawn from the same distribution, $\mathcal{D}$. In the setting of *transfer learning*, however, we would like to apply our trained classifier to examples drawn from a distribution different from the one upon which it was trained. We therefore assume there are two different distributions, $\mathcal{D}^{source}$ and $\mathcal{D}^{target}$, from which data may be drawn. Given this notation we can then precisely state the transfer learning problem as trying to assign labels Y_{test}^{target} to test data X_{test}^{target} drawn from $\mathcal{D}^{target}$, given training data $(X_{train}^{source}, Y_{train}^{source})$ drawn from $\mathcal{D}^{source}$. In this paper we focus on the subproblem of *domain adaptation*, where we assume Y (the set of possible labels) is the same for both D^{source} and D^{target}, while D^{source} and D^{target} themselves are allowed to vary between domains. This is in contrast to the related subproblem of *multi-task learning* [1, 17] in which the marginal distribution of the data is assumed not to change, while the task (and therefore the labels) is allowed to vary from source to target.

In this paper we choose to focus on extensions to the transfer learning setting that allow us to capture some information about D^{target}. One obvious such setting is *inductive transfer learning* where we also provide a few auxiliary labeled data $(X_{auxiliary}^{target}, Y_{auxiliary}^{target})$ from the target domain in addition to the labeled data from the source domain. Due to the presence of labeled target data, this method could also be called *supervised transfer* learning and is the most common setting used by researchers in transfer learning today.

In this work, however, we focus on a new and more challenging paradigm, namely, *transductive transfer learning*, where there is no auxiliary labeled data in the target domain available for training, but where the unlabeled test set on the target domain X_{test}^{target} can be seen during training. Again, due to the lack of labeled target data, this setting could be considered *unsupervised transfer* learning. It is important to point out that *transductive learning* is orthogonal to *transfer learning*. That is, one can have a transductive algorithm that does or does not make the transfer learning assumption, and vice versa. Much of the work in this paper is inspired by the belief that, although distinct, these problems are nevertheless intimately related. More specifically, when trying to solve a transfer problem between two domains, it seems intuitive that looking at the *unlabeled* test data of the target domain during training will improve performance over ignoring this source of information.

We note that the setting of *inductive transfer learning*, in which labeled data from both source and target domains are available for training, serves as a rough upper-bound to the performance of a learner based on *transductive transfer learning*, in which no labeled target data is available. We also considered an additional artificial setting, which we call *relaxed transductive transfer learning*, in our experiments. This setting is almost equivalent to the transductive transfer setting, but the model is allowed to know the proportion of positive examples in the target domain. Although this type of learning is not technically fully unsupervised, in practice estimating this single parameter over the target domain does not require nearly as much labeled target data as learning all the parameters of a fully supervised transfer model, and thus serves as a nice compromise between the two extremes of transduction and supervision.

3 Methods considered

3.1 Maximum entropy models

3.1.1 Inductive learning

Entropy maximization (MaxEnt) [2, 14] is a way of modeling the conditional distribution of labels given examples. Given a set of training examples $X_{train} \equiv \{x_1, \ldots, x_N\}$, their labels $Y_{train} \equiv \{y_1, \ldots, y_N\}$, and the set of features $\mathcal{F} \equiv \{f_1, \ldots, f_F\}$, MaxEnt learns a model consisting of a set of weights corresponding to each class $\Lambda = \{\lambda_{1,y} \ldots \lambda_{F,y}\}_{y \in \{0,1\}}$ over the features so as to maximize the conditional likelihood of the training data, $\mathrm{p}(Y_{train}|X_{train})$, given the model p_Λ. In exponential parametric form, this conditional likelihood can be expressed as:

$$p_\Lambda(y_i = y|x_i) = \frac{1}{Z(x_i)} \exp(\sum_{j=1}^{F} f_j(x_i)\lambda_{j,y}) \qquad (1)$$

where Z is the normalization term. In order to avoid overfitting the training data, these λ's are often further constrained to be near 0 by the use of a regularization term which tries to minimize $\|\Lambda\|_2^2 \equiv \sum_{j,y} (\lambda_{j,y})^2$. Thus the entire expression being optimized is:

$$\underset{\Lambda}{\text{argmax}} \sum_{i=1}^{N} \log\ p_\Lambda(y_i|x_i) - \beta\|\Lambda\|_2^2 \quad (2)$$

where $\beta > 0$ is a parameter controlling the amount of regularization. Maximizing this likelihood is equivalent to constraining the joint expectations of each feature and label in the learned model, $E_\Lambda[f_j, y]$, to match empirical expectations $E_{train}[f_j, y]$ as shown below:

$$E_{train}\left[f_j, y\right] = \frac{1}{N}\sum_{i}^{N} f_j(x_i)\delta_y(y_i) \quad (3)$$

$$E_\Lambda\left[f_j, y\right] = \frac{1}{N}\sum_{i}^{N} f_j(x_i)P_\Lambda(y|x_i) \quad (4)$$

where $\delta_y(y_i) = 1$ if $y = y_i$ and 0 otherwise.

3.1.2 Inductive transfer

Source trained prior models: One recently proposed method [4] for transfer learning in MaxEnt models involves modifying Λ's regularization term. First a model of the source domain, Λ^{source}, is learned by training on $\{X_{train}^{source}, Y_{train}^{source}\}$. Then a model of the target domain is trained over a limited set of labeled target data $\left\{X_{train}^{target}, Y_{train}^{target}\right\}$, but instead of regularizing this Λ^{target} to be near zero by minimizing $\|\Lambda^{target}\|_2^2$, Λ^{target} is instead regularized towards the previously learned source values Λ^{source} by minimizing $\|\Lambda^{target} - \Lambda^{source}\|_2^2$. Thus the modified optimization problem is:

$$\underset{\Lambda^{target}}{\text{argmax}} \sum_{i=1}^{N_{train}^{target}} \log\ p_{\Lambda^{target}}(y_i|x_i) - \beta\|\Lambda^{target} - \Lambda^{source}\|_2^2 \quad (5)$$

where N_{train}^{target} is the number of labeled training examples in the target domain. It should be noted that this model requires Y_{train}^{target} in order to learn Λ^{target} and is therefore a supervised form of *inductive transfer*.

Feature space expansion: Another approach to the problem of inductive transfer learning is explored by Daumé [6, 7]. Here the idea is that there are certain features that are common between different domains, and others that are particular to one or the other. More specifically, we can redefine our feature set $\mathcal{F}$ as being composed of two distinct subsets $\mathcal{F}^{specific} \bigcup \mathcal{F}^{general}$, where the conditional distribution of the features in $\mathcal{F}^{specific}$ differ between X^{source} and X^{target}, while the features in $\mathcal{F}^{general}$ are identically distributed in the source and target. Given this assumption, there is an EM-like algorithm [7] for estimating the parameters of these distributions. The idea is that by expanding the feature space in this way MaxEnt will be able to assign different weights to different versions of the same feature. If a feature is common in both domains its $general$ copy will get most of the weight, while its specific copies (f^{source} and f^{target}) will get less weight, and vice versa.

3.1.3 Transductive transfer: IFT

In this subsection, we present a new approach for the unsupervised setting of transductive transfer learning using MaxEnt. For ease of notation we will use $E^{source}\left[f_j, y\right]$ to mean $E_{x \in \mathcal{D}^{source}}\left[f_j(x), y\right]$, and similarly for $target$.

One problem with transfer in MaxEnt is that the joint distribution of the features with labels differs between the source and target domains. In other words, $E^{source}\left[f_j, y\right]$ does not necessarily equal $E^{target}\left[f_j, y\right]$. If the expectations in the train and test datasets are similar, then the Λ learned on the training data will generalize well to the test data. The more these distributions differ, however, the less well the trained model will perform. Phrased in terms of maximum entropy, we are trying to learn a transformation $G()$ of the feature space $\mathcal{F}$ such that the joint distributions of the source and target features with their labels are aligned:

$$E^{target}\left[G(f_j), y\right] = E^{source}\left[G(f_j), y\right], \forall f_j \in \mathcal{F} \quad (6)$$

One could relax this condition even further by arguing that it is enough to transform only one of the domains, say the source data, so that data from both domains could be separated by a single hyperplane. In maximum entropy phraseology, the relaxed transformation is:

$$E^{target}\left[f_j, y\right] = E^{source}\left[G(f_j), y\right], \forall f_j \in \mathcal{F} \quad (7)$$

The problem with this, of course, is that in the unsupervised transductive transfer case, we do not have Y^{target} and therefore cannot estimate $E^{target}\left[f_j, y\right]$. Hence we approximate $E^{target}\left[f_j, y\right]$ using the joint estimates on the target unlabeled data from a model learned from the source data as shown below:

$$\begin{aligned} E^{target}\left[f_j, y\right] &\approx E_{\Lambda_{source}}^{target}[f_j, y] \\ &= \frac{1}{N_{test}^{target}} \sum_{i=1}^{N_{test}^{target}} f_j(x_i)P_{\Lambda_{source}}(y, x_i) \end{aligned}$$

where N_{test}^{target} is the number of target domain (unlabeled) test examples. These estimates may not reflect the true target expectations, but it is the best we could do in the unsupervised transductive setting. Now we use these expectations to define the source domain transformation G as:

$$\forall_{i=1}^{N_{train}^{source}} \quad G(f_j(x_i)) = f_j \frac{E_{\Lambda_{source}}^{target}[f_j, y_i]}{E^{source}[f_j, y_i]} \quad (8)$$

where $E^{source}[f_j, y_i]$ is given by the formula in (3) and N_{train}^{source} is the number of labeled training data in the source domain. It is easy to show that the empirical feature-label joint expectations of the transformed source data given by $E^{source}[G(f_j, y)]$ defined this way is equal to $E_{\Lambda_{source}}^{target}[f_j, y]$, the model expectations of the original features in the target domain, satisfying the condition in (7). The effect is to rescale $f_j(x)$, putting more weight on features that occur frequently in the target but rarely in the

source (in a conditional sense), and downweighting features that are common in the source but seldom seen in the target. This algorithm can be implemented in an iterative fashion by first training the source model, computing the target expectations using the source model, transforming the source features and then retraining the source model.

In practice, since the target expectation $E^{target}_{\Lambda_{source}}[f_j, y]$ is only approximate, we smooth the transformed features with the original ones in each iteration as follows:

$$G'(f_j(x_i)) = \theta f_j(x_i) + (1-\theta)G(f_j(x_i)) \qquad (9)$$

where θ controls the degree to which we use the target conditional estimates to alter the source conditionals.

3.1.4 Relaxed transductive transfer: biased threshold

A natural way to exploit the known value of the proportion of positive class labels in the target domain is to adjust the decision threshold of the MaxEnt classifier so that the percentage of unlabeled target examples predicted as positive by the source-trained classifier is equal to the known value. We call this intuitive algorithm *biased thresholding*, to reflect the fact that the decision threshold is biased towards the known information on class ratio.

3.2 Support vector machines

3.2.1 Inductive learning: inductive SVMs

Support vector machines (SVMs) [12] take a different approach to the binary classification problem. Instead of explicitly modeling the conditional distribution of the data and using these estimates to predict labels, SVMs try to model the data geometrically. Each example is represented as an F-dimensional real-valued vector of features and is then projected as a point in F-dimensional space. The *inductive SVM* exploits the label information of the training data and fits a discriminative hyperplane between the positively and negatively labeled training examples in this space, so as to best separate the two classes.

3.2.2 Inductive transfer: concatenated data

Recall that in the supervised inductive transfer case, we are given the training sets $(X^{source}_{train}, Y^{source}_{train})$ and $(X^{target}_{train}, Y^{target}_{train})$. Since the SVM does not explicitly model the data distribution, we simply concatenate the source and target labeled data together and provide the entire data for training. The hope is that it will improve on an SVM trained purely on labeled source data, by re-adjusting its hyperplane based on the labeled target data.

3.2.3 Transductive transfer: transductive SVMs

Transduction with SVMs, in contrast to probabilistic models, is quite intuitive. Whereas, in the supervised case, we tried to fit a hyperplane to best separate the labeled training data, in the transductive case, we add in unlabeled testing data which we must also separate. Since we do not know the labels of the testing data, however, we cannot perform a straight forward margin maximization, as in the supervised case. Instead, one can use an iterative algorithm [11] similar in flavor to the MaxEnt iterative feature transformation (IFT) algorithm of section 3.1.3. Specifically, a hyperplane is trained on the labeled source data and then used to classify the unlabeled testing data. As in IFT, one can adjust how confident the hyperplane must be in its prediction in order to use a pseudo-label during the next phase of training (since there are no probabilities, large margin values are used as a measure of confidence). The pseudo-labeled testing data is then, in turn, incorporated in the next round of training. The idea is to iteratively adjust the hyperplane (by switching presumed pseudo-labels) until it is very confident on most of the testing points, while still performing well on the labeled training points.

3.2.4 Relaxed transductive transfer: biased threshold

As with the maximum entropy approaches described in section 3.1.4, transductive SVMs used for transfer can also be adjusted to match the prior proportion of positive examples in the target domain. Specifically, whereas the SVM usually just considers which side of the hyperplane a test example is on in determining its label (i.e., a threshold of 0), this threshold can be moved so that some points that lie nearest on the negative side of the hyperplane and would normally be given a negative label, would instead receive a positive one, or vice verse.

4 Investigation

4.1 Domain

We now turn to *protein name recognition*, an interesting problem domain [15, 19, 10] in which to test these methods. In this setting you are given text related to biological research (usually abstracts, captions, and full body text from biological journal articles) which is known to contain mentions of protein names. The goal is to identify which words are part of a protein name mention, and which are not. One major difficulty is that there is a large variance in how these proteins are mentioned and annotated between different authors, journals, and sub-disciplines of biology. Because of this variance it is often difficult to collect a large corpus of truly identically distributed training examples. Instead, researchers are often faced with heterogeneous sources of data, both for training and testing, thus violating one of the key assumptions of most standard machine learning algorithms and indicating a need for a transfer learning approach.

Table 1. Summary of data used in experiments

Corpus name (Abbr.)	Abstracts	Tokens	% Positive
UTexas (UT)	748	216,795	6.6%
Yapex (Y)	200	60,530	15.0%
Yapex-train (YTR)	160	48,417	15.1%
Yapex-test (YTT)	40	12,113	14.5%

4.2 Data and evaluation

Our corpora are abstracts from biological journals coming from two sources: University of Texas, Austin (UT) [3] and Yapex [8]. Each abstract was tokenized and each token was hand-labeled as either being part of a protein name or not. We used a standard natural language toolkit [5] to compute tens of thousands of binary features on each of these tokens, encoding such information as capitalization patterns and contextual information of surrounding words.

Some summary statistics for these data are shown in table 1. We purposely chose corpora that differed in two important dimensions: the total amount of data collected and the relative proportion of positively labeled examples in each dataset. For all our experiments, we used the larger UT dataset as our source domain and the smaller Yapex dataset as our target. We also split the Yapex data into two parts: *Yapex-train* (YTR) consisting of 80% of the data, and *Yapex-test* (YTT), consisting of the remaining 20%.

Because of the relatively small proportion of positive examples in both the UT and Yapex datasets, we are more interested in achieving both high precision and recall of protein name mentions instead of simply maximizing classification accuracy and thus use the $F1$ measure, which combines precision and recall into one metric, as our main evaluation measure.

4.3 Experiments and results

Table 2 summarizes the relative performance of the various methods (cf. section 3) in four different learning settings (cf. section 2). The inductive experiment is dominated by MaxEnt's 82% F1 compared to TSVM's 73%. Moving to the transductive transfer setting causes both methods' performances to fall, but MaxEnt falls most sharply, causing it to lose its entire lead over TSVM. TSVM is able to adjust its hyperplane in light of the transfer test data and stabilize its performance at 60%, even though it is unlabeled, because it knows where these points lie relative to the labeled training points in feature space. Similarly, we see the effect of our iterative feature transformation algorithm (*IFT*, section 3.1.3) on MaxEnt's transductive transfer performance. Indeed, iteratively combining the approximate joint feature-label expectations in the target data with the true joints of the source data improves the overall performance on the target data. It seems this method is bounded, however, by the quality of the initial target labels generated by the source-trained classifier.

In the relaxed transductive transfer setting, where the target dataset is still unlabeled but all algorithms are told the expected proportion of positive examples, TSVM excels. Again, while MaxEnt is able to make significant use of this information (note the jump to 67% F1 from 54%), it seems TSVM does a better job leveraging the prior knowledge into better performance.

Finally, the last column of table 2 compares the performance of the methods for inductive transfer learning: *Regularize* and *Expand*, both described in section 3.1.2. We can see that both methods handily outperform the transductive transfer methods described in the second column of table 2, and for the most part outperform even the relaxed transductive transfer versions in column three. This should not be surprising given the fact that the inductive transfer methods can actually see some labeled examples from the target domain and thus better estimate the conditional expectation of the features in the target data. Likewise they can also assess the proportion of positive examples and adjust their decision functions accordingly. It is surprising, however, that these methods do not significantly outperform the inductive learning methods described in the first column of table 2. This suggests that these inductive transfer methods are relying almost entirely on their labeled target data to train their classifiers, and are not making full use of the large amount of labeled source data. The regularized maximum entropy model does outperform the basic MaxEnt in the inductive setting, but not by as much as might have been hoped for.

In order to measure how much these inductive transfer methods' explicit modeling of the transfer problem was responsible for their performance, we compared them to the baselines of ISVM, TSVM, and MaxEnt trained on a simple concatenation of the labeled source and target training data. These transfer-agnostic methods clearly benefited from the addition of labeled target data (as compared to column *TransductiveTransfer*), yet still yielded consistently lower F1 than the transfer-aware *Regularize* and *Expand* methods, suggesting that the mere presence of labeled sets of both types (source and target) of data is not enough to account for the transfer methods' superior results. Instead, it seems it is the modeling of the different domains in the transfer problem, even in simple ways, that provides the extra boost to performance.

5 Conclusions & future work

We have seen that even a small amount of prior knowledge about the target domain can greatly improve performance in a transductive transfer problem. We also notice that even large amounts of source data cannot overcome the

Table 2. Summary of % accuracy (**Acc**), precision (**Prec**), recall (**Rec**), and F1 for regular maximum entropy (**Basic**), Iterative Feature Transformation MaxEnt (**IFT**), prior-based regularized MaxEnt (**Regularize**), and feature expansion MaxEnt (**Expand**), inductive SVM (**ISVM**), and transductive SVM (**TSVM**) models under the conditions of classic inductive learning, (**Induction**), unsupervised transductive transfer learning, (**TransductTransfer**), relaxed transductive transfer, (**RelaxTransductTransfer**), and supervised inductive transfer (**InductTransfer**).

Method	Induction				TransductTransfer				RelaxTransductTransfer				InductTransfer			
	Acc	Prec	Rec	**F1**	Acc	Prec	Rec	**F1**	Acc	Prec	Rec	**F1**	Acc	Prec	Rec	**F1**
MAXIMUM ENTROPY																
Basic	95	85	78	**82**	89	75	42	**54**	90	65	68	**67**	91	81	54	**65**
IFT, 1 iter	-	-	-	-	79	41	90	**56**	-	-	-	-	-	-	-	-
IFT, 2 iters	-	-	-	-	82	45	86	**59**	-	-	-	-	-	-	-	-
Regularize	-	-	-	-	-	-	-	-	-	-	-	-	96	87	84	**85**
Expand	-	-	-	-	-	-	-	-	-	-	-	-	93	84	62	**72**
SUPPORT VECTOR MACHINES																
ISVM	92	78	58	**67**	90	86	40	**54**	90	86	40	**55**	92	86	52	**65**
TSVM	92	68	79	**73**	91	86	46	**60**	92	72	75	**73**	93	86	58	**70**

advantage of having access to labeled data drawn from the target distribution. We have also seen the degree to which pseudo-labeling based schemes (in both TSVM's margin-based model and our MaxEnt's IFT-based model) can improve performance by incorporating the unlabeled structure of the target domain. Finally we have seen that, while both the MaxEnt and SVM models perform well in the transductive setting, the margin based SVM seems to adapt better to the unlabeled data.

In the future, we would like to further investigate the theoretical properties of the IFT-type algorithms while extending these methods to use sequential, rather than simply binary, classifiers like conditional random fields [17].

References

[1] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. In *JMLR 6*, pages 1817 – 1853, 2005.

[2] A. L. Berger, S. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani, and Y. Wong. Comparative experiments on learning information extractors for proteins and their interactions. In *Journal of AI in Medicine*, 2004. Data from *ftp://ftp.cs.utexas.edu/pub/mooney/bio-data/proteins.tar.gz*.

[4] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. In D. Lin and D. Wu, editors, *EMNLP 2004*, pages 285–292. ACL, 2004.

[5] W. W. Cohen. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. http://minorthird.sourceforge.net, 2004.

[6] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[7] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. In *Journal of Artificial Intelligence Research 26*, pages 101–126, 2006.

[8] K. Franzén, G. Eriksson, F. Olsson, L. Asker, P. Lidn, and J. Cöster. Protein names and how to find them. In *International Journal of Medical Informatics*, 2002.

[9] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *CAP*, Nice, France, 2005.

[10] K. Ji, M. Ohta, and Y. Tsujii. Tuning support vector machines for biomedical named entity recognition. In *ACL Workshop on Natural Language Processing in the Biomedical Domain.*, 2002.

[11] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML 16*, 1999.

[12] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer, 2002.

[13] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML*, 2003.

[14] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification, 1999.

[15] L. Shi and F. Campagne. Building a protein name dictionary from full text: a machine learning term extraction approach. In *BMC Bioinformatics 6:88*, 2005.

[16] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831. ACM, 2005.

[17] C. Sutton and A. McCallum. Composition of conditional random fields for transfer learning. In *HLT/EMLNLP*, 2005.

[18] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[19] R. C. Wang, A. Tomasic, R. E. Frederking, and W. W. Cohen. Learning to extract gene-protein names from weakly-labeled text in preparation. In *preparation*, 2006.

[20] X. Zhu. Semi-supervised learning literature survey. In *Technical Report 1530*. University of Wisconsin, 2005.