Biased Representation Learning for Domain Adaptation

Fei Huang , Alexander Yates
Temple University
Computer and Information Sciences
324 Wachman Hall
Philadelphia, PA 19122
{fhuang,yates}@temple.edu

Abstract

Representation learning is a promising technique for discovering features that allow supervised classifiers to generalize from a source domain dataset to arbitrary new domains. We present a novel, formal statement of the representation learning task. We argue that because the task is computationally intractable in general, it is important for a representation learner to be able to incorporate expert knowledge during its search for helpful features. Leveraging the Posterior Regularization framework, we develop an architecture for incorporating biases into representation learning. We investigate three types of biases, and experiments on two domain adaptation tasks show that our biased learners identify significantly better sets of features than unbiased learners, resulting in a relative reduction in error of more than 16% for both tasks, with respect to existing state-of-the-art representation learning techniques.

1 Introduction

Supervised natural language processing (NLP) systems have been widely used and have achieved impressive performance on many NLP tasks. However, they exhibit a significant drop-off in performance when tested on domains that differ from their training domains. (Gildea, 2001; Sekine, 1997; Pradhan et al., 2007) One major cause for poor performance on out of-domain texts is the traditional representation used by supervised NLP systems (Ben-David et al., 2007). Most systems depend on lexical features, which can differ greatly between domains, so that important words in the test data may never be seen in the training data. The connection between words and labels may also change across domains. For instance, "signaling" appears only as a present participle (VBG) in WSJ text (as in, "signaling that..."), but predominantly as a noun (as in "signaling pathway") in biomedical text.

Recently, several authors have found that learning new features based on distributional similarity can significantly improve domain adaptation (Blitzer et al., 2006; Huang and Yates, 2009; Turian et al., 2010; Dhillon et al., 2011). This framework is attractive for several reasons: experimentally, learned features can yield significant improvements over standard supervised models on out-of-domain tests. Moreover, since the representation-learning techniques are unsupervised, they can easily be applied to arbitrary new domains. There is no need to supply additional labeled examples for each new domain.

Traditional representations still hold one significant advantage over representation-learning, however: because features are hand-crafted, these representations can readily incorporate the linguistic or domain expert knowledge that leads to state-of-the-art in-domain performance. In contrast, the only guide for existing representation-learning techniques is a corpus of unlabeled text.

To address this shortcoming, we introduce representation-learning techniques that incorporate a domain expert's preferences over the learned features. For example, out of the set of all possible distributional-similarity features, we might prefer those that help predict the labels in a labeled training data set. To capture this preference, we might bias a representation-learning algorithm towards features with low joint entropy with the labels in the training data. This particular biased form of

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1313–1323, Jeju Island, Korea, 12–14 July 2012.

representation learning is a type of semi-supervised learning that allows our system to learn *task-specific* representations from a source domain's training data, rather than the single representation for all tasks produced by current, unsupervised representation-learning techniques.

We present a novel formal statement of representation learning, and demonstrate that it is computationally intractable in general. It is therefore critical for representation learning to be flexible enough to incorporate the intuitions and knowledge of human experts, to guide the search for representations efficiently and effectively. Leveraging the Posterior Regularization framework (Ganchev et al., 2010), we present an architecture for learning representations for sequence-labeling tasks that allows for biases. In addition to a bias towards task-specific representations, we investigate a bias towards representations that have similar features across domains, to improve domain-independence; and a bias towards multi-dimensional representations, where different dimensions are independent of one another. In this paper, we focus on incorporating the biases with HMM-type representations (Hidden Markov Model). However, this technique can also be applied to other graphical model-based representations with little modification. Our experiments show that on two different domain-adaptation tasks, our biased representations improve significantly over unbiased ones. In a part-of-speech tagging experiment, our best model provides a 25% relative reduction in error over a state-of-the-art Chinese POS tagger, and a 19% relative reduction in error over an unbiased representation from previous work.

The next section describes background and previous work. Section 3 introduces our framework for learning biased representations. Section 4 describes how we estimate parameters for the biased objective functions efficiently. Section 5 details our experiments and results, and section 6 concludes and outlines directions for future work.

2 Background and Previous Work

2.1 Terminology and Notation

A *representation* is a set of features that describe data points. Formally, given an instance set $\mathcal{X}$, it is a function $\mathcal{R} : \mathcal{X} \rightarrow \mathcal{Y}$ for some suitable space $\mathcal{Y}$ (often $\mathbf{R}^d$), which is then used as the input space for a classifier. For instance, a traditional representation for POS tagging over vocabulary V would include (in part) $|V|$ dimensions, and would map a word to a binary vector with a 1 in only one of the dimensions. By a *structured representation*, we mean a function R that incorporates some form of joint inference. In this paper, we use Viterbi decoding of variants of Hidden Markov Models (HMMs) for our structured representations, although our techniques are applicable to arbitrary (Dynamic) Bayes Nets. A *domain* is a probability distribution D over the instance set $\mathcal{X}$; $R(D)$ denotes the induced distribution over $\mathcal{Y}$. In *domain adaptation* tasks, a learner is given samples from a *source* domain D_S, and is evaluated on samples from a *target* domain D_T.

2.2 Theoretical Background

Ben-David *et al.* (2010) give a theoretical analysis of domain adaptation which shows that the choice of representation is crucial. A good choice is one that minimizes error on the training data, but equally important is that the representation must make data from the two domains look similar. Ben-David *et al.* show that for every hypothesis h, we can provably bound the error of h on the target domain by its error on the source domain plus a measure of the distance between D_S and D_T:

$$\mathrm{E}_{x\sim D_T}\mathcal{L}(x, R, f, h) \leq \mathrm{E}_{x\sim D_S}\mathcal{L}(x, R, f, h) + d_1(R(D_S), R(D_T))$$

where $\mathcal{L}$ is a loss function, f is the target function, and the variation divergence d_1 is given by

$$d_1(D, D') = 2 \sup_{B\in\mathcal{B}} |Pr_D[B] - Pr_{D'}[B]| \quad (1)$$

where $\mathcal{B}$ is the set of measurable sets under D, D'.

2.3 Problem Formulation

Ben-David *et al.*'s theory provides learning bounds for domain adaptation under a fixed R. We now reformulate this theory to define the task of representation learning for domain adaptation as the following optimization problem: given a set of unlabeled instances U_S drawn from the source domain and unlabeled instances U_T from the target domain, as well as a set of labeled instances L_S drawn from

the source domain, identify a function R^* from the space of possible representations $\mathcal{R}$:

$$R^* = \underset{R \in \mathcal{R}}{\operatorname{argmin}} \{ \min_{h \in \mathcal{H}} (\mathrm{E}_{x \sim D_S} \mathcal{L}(x, R, f, h)) + d_1(R(D_S), R(D_T)) \} \quad (2)$$

Unlike most learning problems, where the representation R is fixed, this problem formulation involves a search over the space of representations and hypotheses. The equation also highlights an important underlying tension: the best representation for the source domain would naturally include domain-specific features, and allow a hypothesis to learn domain-specific patterns. We are aiming, however, for the best *general* classifier, that happens to be trained on training data from one or a few domains. Domain-specific features would contribute to distance between domains, and to classifier errors on data taken from unseen domains. By optimizing for this combined objective function, we allow the optimization method to trade off between features that are best for classifying source-domain data and features that allow generalization to new domains.

Naturally, the objective function in Equation 2 is completely intractable. Just finding the optimal hypothesis for a fixed representation of the training data is intractable for many hypothesis classes. And the d_1 metric is intractable to compute from samples of a distribution, although Ben-David *et al.* propose some tractable bounds (2007; 2010). We view Equation 2 as a high-level goal rather than a computable objective. We leverage prior knowledge to bias the representation learner towards attractive regions of the representations space $\mathcal{R}$, and we develop efficient, greedy optimization techniques for learning effective representations.

2.4 Previous Work

There is a long tradition of research on representations for NLP, mostly falling into one of three categories: 1) vector space models and dimensionality reduction techniques (Salton and McGill, 1983; Turney and Pantel, 2010; Sahlgren, 2005; Deerwester et al., 1990; Honkela, 1997) 2) using structured representations to identify clusters based on distributional similarity, and using those clusters as features (Lin and Wu, 2009; Candito and Crabbé, 2009; Huang and Yates, 2009; Ahuja and Downey, 2010; Turian et al., 2010; Huang et al., 2011); 3) and structured representations that induce multi-dimensional real-valued features (Dhillon et al., 2011; Emami et al., 2003; Morin and Bengio, 2005). Our work falls into the second category, but builds on the previous work by demonstrating how to improve the distributional-similarity clusters with prior knowledge. To our knowledge, we are the first to apply semi-supervised representation learning techniques for structured NLP tasks.

Most previous work on domain adaptation has focused on the case where some labeled data is available in both the source and target domains (Daumé III, 2007; Jiang and Zhai, 2007; Daumé III et al., 2010). Learning bounds are known (Blitzer et al., 2007; Mansour et al., 2009). A few authors have considered domain adaptation with no labeled data from the target domain (Blitzer et al., 2006; Huang et al., 2011) by using features based on distributional similarity. We demonstrate empirically that incorporating biases into this type of representation-learning process can significantly improve results.

3 Biased Representation Learning

As before, let U_S and U_T be unlabeled data, and L_S be labeled data from the source domain only. Previous work on representation learning with Hidden Markov Models (HMMs) (Huang and Yates, 2009) has estimated parameters θ for the HMM from unlabeled data alone, and then determined the Viterbi-optimal latent states for training and test data to produce new features for a supervised classifier. The objective function for HMM learning in this case is marginal log-likelihood, optimized using the Baum-Welch algorithm:

$$\mathcal{L}(\theta) = \sum_{\mathbf{x} \in U_S \cup U_T} \log \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{Y} = \mathbf{y} | \theta) \quad (3)$$

where $\mathbf{x}$ is a sentence, $\mathbf{Y}$ is the sequence of latent random variables for the sentence, and $\mathbf{y}$ is an instance of the latent sequence. The joint distribution in an HMM factors into observation and transition distributions, typically mixtures of multinomials:

$$p(\mathbf{x}, \mathbf{y} | \theta) = P(y_1) P(x_1 | y_1) \prod_{i \geq 2} P(y_i | y_{i-1}) P(x_i | y_i)$$

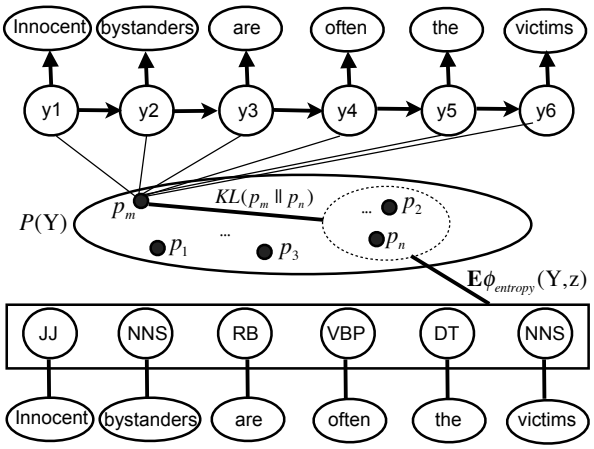

Figure 1: Illustration of how the entropy bias is incorporated into HMM learning. The dotted oval shows the space of desired distributions in the hidden space, which have small or zero entropy with the real labels. The learning algorithm aims to maximize the log-likelihood of the unlabeled data, and to minimize the KL divergence between the real distribution, p_m, and the closest desired distribution, p_n.

Intuitively, this form of representation learning identifies clusters of distributionally-similar words: those words with the same Viterbi-optimal latent state. The Viterbi-optimal latent states are then used as features for the supervised classifier. Our previous work (2009) has shown that the features from the learned HMM significantly improve the accuracy of POS taggers and chunkers on benchmark domain adaptation datasets.

We use the HMM model from our previous work (2009) as our baseline. Our techniques follow the same general setup, as it provides an efficient and empirically-proven starting point for exploring (one part of) the space of possible representations. Note, however, that the HMM on its own does not provide even an approximate solution to the objective function in our problem formulation (Eqn. 2), since it makes no attempt to find the representation that minimizes loss on labeled data. To address this and other concerns, we modify the objective function for HMM training. Specifically, we encode biases for representation learning by defining a set of *properties* ϕ that we believe a good representation function would minimize. One possible bias is that the HMM states should be predictive of the labels in labeled training data. We can encode this as a property that computes the entropy between the HMM states and the labels. For example, in Figure 1, we want to learn the best HMM distribution for the sentence "Innocent bystanders are often the victims" for POS tagging task. The hidden sequence $y_1, y_2, y_3, y_4, y_5, y_6$ can have any distribution $p_1, p_2, p_3, ..., p_m, ..., p_n$ from the latent space $\mathcal{Y}$. Since we are doing POS tagging, we want the distribution to learn the information encoded in the original POS labels "JJ NNS RB VBP DT NNS". Therefore, by calculating the entropy between the hidden sequence and real labels, we can identify a subset of desired distributions that have low entropy, shown in the dotted oval. By minimizing the KL divergence between the learned distribution and the set of desired distributions, we can find the best distribution which is the closest to our desire.

The following subsections describe the specific properties we investigate; here we show how to incorporate them into the objective function. Let $\mathbf{z}$ be the sequence of labels in L_S, and let $\phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ be a property of the completed data that we wish the learned representation to minimize, based on our prior beliefs. Let Q be the subspace of the possible distributions over $\mathbf{Y}$ that have a small expected value for ϕ: $Q = \{q(\mathbf{Y}) | \mathbf{E}_{\mathbf{Y}\sim q}[\phi(\mathbf{x}, \mathbf{Y}, \mathbf{z})] \leq \xi\}$, for some constant ξ. We then add penalty terms to the objective function (3) for the divergence between the HMM distribution p and the "good" distributions q, as well as for ξ:

$$\mathcal{L}(\theta) - \min_{q,\xi} [\mathbf{KL}(q(\mathbf{Y}) || p(\mathbf{Y}|\mathbf{x}, \theta)) + \sigma|\xi|] \quad (4)$$

$$\text{s.t. } \mathbf{E}_{\mathbf{Y}\sim q}[\phi(\mathbf{x}, \mathbf{Y}, \mathbf{z})] \leq \xi \quad (5)$$

where $\mathbf{KL}$ is the Kullback-Leibler divergence, and σ is a free parameter indicating how important the bias is compared with the marginal log likelihood.

To incorporate multiple biases, we define a vector of properties ϕ, and we constrain each property $\phi_i \leq \xi_i$. Everything else remains the same, except that in the penalty term $\sigma|\xi|$, the absolute value is replaced with a suitable norm: $\sigma \|\xi\|$. To allow ourselves to place weights on the relative importance of the different biases, we use a norm of the form $\|\mathbf{x}\|_A = \sqrt{(\mathbf{x}^t A \mathbf{x})}$, where A is a diagonal matrix whose diagonal entries A_{ii} are free parameters that provide weights on the different properties. For our

experiments, we set the free parameters σ and A_{ii} using a grid search over development data, as described in Section 5.[1]

3.1 A Bias for Task-specific Representations

Current representation learning techniques are unsupervised, so they will generate the exact same representation for different tasks. Yet it is exceedingly rare that two state-of-the-art NLP systems for different tasks share the same feature set, even if they do tend to share some core set of lexical features.

Traditional non-learned (*i.e.*, manually-engineered) representations essentially always include task-specific features. In response, we propose to bias our representation learning such that the learned representations are optimized for a specific task. In particular, we propose a property that measures how difficult it is to predict the labels in training data, given the learned latent states. Our *entropy property* uses conditional entropy of the labels given the latent state as the measure of unpredictability:

$$\phi^{entropy}(\mathbf{y}, \mathbf{z}) = -\sum_i \tilde{P}(y_i, z_i) \log \tilde{P}(z_i|y_i) \quad (6)$$

where $\tilde{P}$ is the empirical probability and i indicates the ith position in the data. We can plug this feature into Equation 5 to obtain a new version of Equation 4 as an objective function for task-specific representations. We refer to this model as **HMM+E**. Unlike previous formulations for supervised and semi-supervised dimensionality reduction (Zhang et al., 2007; Yang et al., 2006), our framework works efficiently for structured representations.

3.2 A Bias for Domain-Independent Features

Following the theory in Section 2.2, we devise a biased objective to provide an explicit mechanism for minimizing the distance between the source and target domain. As before, we construct a property of the completed data:

$$\phi^{distance}(\mathbf{y}) = d_1(\tilde{P}_S, \tilde{P}_T)$$

where $\tilde{P}_S(Y)$ is the empirical distribution over latent state values estimated from source-domain latent states, and similarly for $\tilde{P}_T(Y)$. Essentially, minimizing this property will bias the the representation towards features that appear approximately as often in the source domain as the target domain. We refer to the model trained with a bias of minimizing $\phi^{distance}$ as **HMM+D**, and the model with both $\phi^{distance}$ and $\phi^{entropy}$ biases as **HMM+D+E**.

[1]Note that ξ, unlike A and σ, is not a free parameter. It is explicitly minimized in the modified objective function.

3.3 A Bias for Multi-Dimensional Representations

Words are multidimensional objects. In English, words can be nouns or verbs, singular or plural, count or mass, just to name a few dimensions along which they may vary. Factorial HMMs (FHMMs) (Ghahramani and Jordan, 1997) can learn multi-dimensional models, but inference and learning are complex and computationally expensive even in supervised settings. Our previous work (2010) created a multi-dimensional representation called an "I-HMM" by training several HMM layers independently; we showed that by finding several latent categories for each word, this representation can provide useful and domain-independent features for supervised learners. In this work, we also learn a similar multi-dimensional model (**I-HMM+D+E**), but within each layer we add in the two biases described above. While more efficient than FHMMs, the drawback of these I-HMM-based models is that there is no mechanism to encourage the different HMM models to learn different things. As a result, the layers may produce similar or equivalent features describing the dominant aspect of distributional similarity in the data, but miss features that are less strong, but still important, in the data.

To encourage learning a truly multi-dimensional representation, we add a bias towards I-HMM models in which each layer is different from all previous layers. We define an entropy-based *predictability* property that measures how predictable each previous layer is, given the current one. Formally, let y_i^l denote the hidden state at the ith position in layer l of the model. For a given layer l, this property measures the conditional entropy of $\mathbf{y}_m$ given $\mathbf{y}_l$, summed over layers $m < l$, and subtracts this from the maximum possible entropy:

$$\phi_l^{predict}(\mathbf{y}) = MAX + \sum_{i;m<l} \tilde{P}(y_i^l, y_i^m) \log \tilde{P}(y_i^m|y_i^l)$$

The entropy between layer l and the previous layer-

s m measures how unpredictable the previous layers are, given layer l. By biasing the model such that MAX minus the entropy approaches zero, we encourage layer l towards completely different features from previous layers. We call the model with this bias **P-HMM+D+E**.

4 Efficient Parameter Estimation

Several machine learning paradigms have been developed recently for incorporating biases and constraints into parameter estimation (Liang et al., 2009; Chang et al., 2007; Mann and McCallum, 2007). We leverage the Posterior Regularization (PR) framework for our problem because of its flexibility in handling different kinds of biases; we provide a brief overview of the technique here, but see (Ganchev et al., 2010) for full details.

4.1 Overview of PR

PR introduces a modified EM algorithm to handle constrained objectives, like Equation 4. The modified E-step estimates a distribution $q(\mathbf{Y})$ that is close to the current estimate of $p(\mathbf{Y}|\mathbf{x}, \theta)$, but also close to the ideal set of distributions that (in expectation) have $\phi = 0$ for each property ϕ. The M step remains the same, except that it re-estimates parameters with respect to expected latent states computed with q rather than p.

E step:

$$q^{t+1} = \arg\min_{q} \min_{\xi} \mathbf{KL}(q(\mathbf{Y})||p(\mathbf{Y}|\mathbf{x}, \theta^t)) + \sigma \|\xi\|$$
$$\text{s.t. } \mathbf{E}_q[\phi(\mathbf{x}, \mathbf{Y}, \mathbf{z})] \leq \xi$$

M step:

$$\theta^{t+1} = \operatorname{argmax}_{\theta} \mathbf{E}_{q^{t+1}}[\log p(\mathbf{x}, \mathbf{Y}|\theta^t))]$$

To make the optimization task in the E-step more tractable, PR transforms it to a dual problem:

$$\max_{\lambda \geq 0, \|\lambda\|_* \leq \sigma} -\log \sum_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{x}, \theta) \exp\{-\lambda \cdot \phi(\mathbf{x}, \mathbf{Y}, \mathbf{z})\}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. The gradient of this dual objective is $-\mathbf{E}_q[\phi(\mathbf{x}, \mathbf{Y}, \mathbf{z})]$. A projected subgradient descent algorithm is used to perform the optimization.

4.2 Modifying ϕ for Tractability

In unstructured settings, this optimization problem is relatively straightforward. However, for structured representations, we need to ensure that the dynamic programming algorithms needed for inference remain tractable for the biased objectives. For efficient PR over structured models, the properties ϕ need to be decomposed as a sum over the cliques in the structured model. Unfortunately, the properties we mention above do not decompose so nicely, so we must resort to approximations.

In order to efficiently compute the expected value of the entropy property with respect to $\mathbf{Y} \sim q$, we need to be able to compute each component $\mathbf{E}_{Y_i \sim q}[\phi^{entropy}(Y_i, z_i)]$ separately. Yet $\tilde{P}$ depends on the setting of other latent states Y_j in the corpus. To avoid this problem, we pre-compute the expected empirical distributions over the completed data. For each specific value y and z:

$$\tilde{P}_q(y, z) = \frac{1}{|L_S|} \sum_{x} \sum_{i=1}^{|x|} \mathbf{1}[z_i = z] q(Y_i = y)$$

$$\tilde{P}_q(y) = \frac{1}{|L_S|} \sum_{x} \sum_{i=1}^{|x|} q(Y_i = y)$$

These expected empirical distributions $\tilde{P}_q$ can be computed efficiently using standard inference algorithms, such as the forward algorithm for HMMs. Note that $\tilde{P}_q$ depends on q, but unlike the original $\tilde{P}$ from Equation 6, they do not depend on the data completions $\mathbf{y}$. Thus we can compute $\tilde{P}_q$ once for each q^t, and then substitute it for $\tilde{P}$ for all values of $\mathbf{Y}$ in the computation of $\mathbf{E}_{\mathbf{Y} \sim q} \phi^{entropy}(\mathbf{Y}, \mathbf{z})$, making this computation tractable. For the entropy-based predictability properties, the calculation is similar, but instead of using the label z, we use the decoded states y_i^l from previous layers.

For the distance property, Ben-David *et al.*'s analysis depends on a particular notion of distance (Eqn. 1) that is computationally intractable. They also propose more tractable lower bounds, but these are again incompatible with the PR framework. Since no computationally feasible exact algorithm exists for this distance feature, we resort to a crude but efficient approximation of this measure: for each pos-

sible value y of the latent states, we define:

$$\phi_y^{dist}(\mathbf{y}) = \sum_{i|x_i \in U_S} \frac{\mathbf{1}[y_i = y]q(Y_i = y)}{|U_S|} - \sum_{i|x_i \in U_T} \frac{\mathbf{1}[y_i = y]q(Y_i = y)}{|U_T|}$$

Each of these individual properties is tractable for structured models. Combining these properties using the $\|\cdot\|_A$ norm results in a Euclidean distance (weighted by A) between the frequencies of features in each domain, rather than d_1 distance.

5 Experiments

We tested the structured representations with biases on two NLP tasks: Chinese POS tagging and English NER. In both cases, we use a domain adaptation setting where no labeled data is available for the target domain — a particularly difficult setting, but one that provides a strong test for an NLP system's ability to generalize . In our work (Huang and Yates, 2009), we used a plain HMM for domain adaptation tasks in which there is labeled source data and unlabeled source and target data, but no labeled target data for training. Therefore, here, we use the HMM technique as a baseline, and build on it by including biases.

5.1 Chinese POS tagging

We use the UCLA Corpus of Written Chinese, which is a part of The Lancaster Corpus of Mandarin Chinese (LCMC). The UCLA Corpus consists of 11,192 sentences of word-segmented and POS-tagged text in 13 genres. We use gold-standard word segmentation labels during training and testing. The LCMC tagset consists of 50 Chinese POS tags. Each genre averages 5284 word tokens, for a total of 68,695 tokens among all genres. We use the 'news' genre as our source domain and randomly select 20% of every other genre as labeled test data. To train our representation models, we use the 'news' text, plus the remaining 80% of the texts from the other genres. We use 90% of the labeled news text for training, and 10% for development. We replace *hapax legomena* in the unlabeled data with the special symbol *UNKNOWN*, and also do the same for word types in the labeled test sets that never appear in our unlabeled training texts.

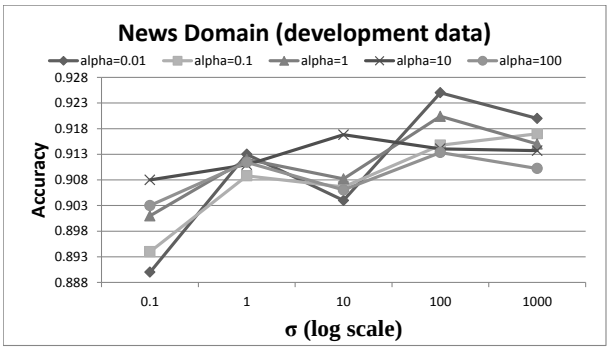

Figure 2: Grid search for parameters on news text

Following our previous HMM setup in (Huang and Yates, 2009) for consistency, we use an HMM with 80 latent states. For our multi-layer models, we use 7 layers of HMMs. We tuned the free parameters σ and A on development data. We varied σ from 0.1 to 1000. To tune A, we start by setting the diagonal entry for $\phi^{entropy}$ to 1, without loss of generality. We then tie all the entries in A for ϕ_y^{dist} to a single parameter α, and tie all of the entries for $\phi_y^{predict}$ to a parameter β. We vary α and β over the set $\{0.01,0.1,1,10,100\}$. Figure 2 shows our results for σ and α on news development data. A setting of $\alpha = 0.01$ and $\sigma = 100$ performs best, with all $\sigma = 100$ doing reasonably well. Results for each of these models on the general fiction test text confirm the general trends seen on development data — a comforting sign, since this indicates we can optimize the free parameters on in-domain development data, rather than requiring labeled data from the target domain. Our models tended to perform better with increasing β on development data, though with diminishing returns. We pick the largest setting tested, $\beta = 100$, for our final models.

We use a linear-chain Conditional Random Field (CRF) for our supervised classifier. To incorporate the learned representations, we use the Viterbi Algorithm to find the optimal latent state sequence from each HMM-based model and then use the optimal states as features in the CRF. Table 1 presents the full list of features in the CRF. To handle Chinese, we add in two features introduced in previous work (Wang et al., 2009): radical features and repeated characters. A *radical* is a portion of a Chinese character that consists of a small number of pen or brush strokes in a regular pattern.

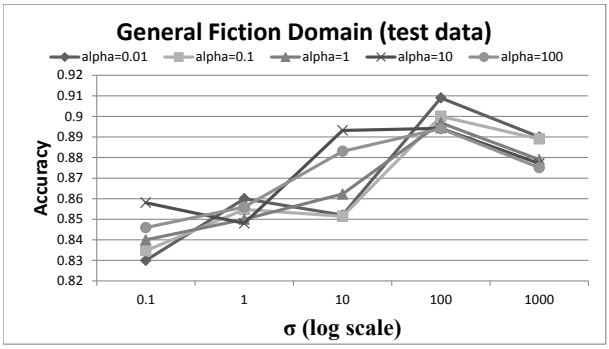

Figure 3: Validating parameter settings on fiction text

CRF Feature Set
Transition
$\forall_z \mathbf{1}[z_j = z]$
$\forall_{z,z'} \mathbf{1}[z_j = z \text{ and } z_{j-1} = z']$
Word
$\forall_{w,z} \mathbf{1}[x_j = w \text{ and } z_j = z]$
Radical
$\forall_{z,r} \mathbf{1}[\exists_{c \in x_j} radical(c) = r \text{ and } z_j = z]$
Repeated Words
$\forall_{A,B,z} \mathbf{1}[x_j = AABB \text{ and } z_j = z]$
$\forall_{A,z} \mathbf{1}[(x_j = AA\text{的or } x_j = AA\text{地}) \text{ and } z_j = z]$
$\forall_{A,B,z} \mathbf{1}[x_j = ABAB \text{ and } z_j = z]$
Features from Representation Learning
$\forall_{y,l,z} \mathbf{1}[y^l_j = y \text{ and } z_j = z]$

Table 1: **Features used in our Chinese POS tagging CRF systems.** c represents a character within a word.

Table 2 shows our results. We compare against the Baseline CRF without any additional representations and the unbiased HMM, a state-of-the-art domain adaptation technique from previous work, over all 13 domains (source and target). We also compare against a state-of-the-art Chinese POS tagger for in-domain text, the CRF-based Stanford tagger (Tseng et al., 2005), retrained for this corpus. H-MM+D+E outperforms the Stanford tagger on 10 out of 12 target domains and the unbiased HMM on all domains, while the P-HMM+D+E outperforms the Stanford tagger (2.6% average improvement) and HMM (1.7%) on all 12 target domains. The I-HMM+D+E is slightly better than the HMM+D+E (.3%), but incorporating the multi-dimensional bias (P-HMM+D+E) adds an additional 0.6% improvement.

Our interpretation for the success of I-HMM+D+E and P-HMM+D+E is that the increase in the state space of the models yields improved performance. Because P-HMM+D+E biases against redundant states found in I-HMM+D+E, it effectively increases the state space beyond I-HMM+D+E. Ahuja and Downey (2010) and our own work with HMMs as representations (2010) have previously shown that increasing the state space of the HMM can significantly improve the representation, but memory constraints eventually prevent further progress this way. The I-HMM+D+E and P-HMM+D+E models can provide similar benefits, but because they split parameters across multiple HMMs, they can accommodate much greater state spaces in the same amount of memory.

We also tested the entropy and distance biases separately. Figure 4 shows the result of the distance-biased HMM+D on the general-fiction test text, as we vary σ over the set $\{0.1,1,10,100,1000\}$ (we observed similar results for other domains). For all values of σ, the biased representation outperforms the unbiased HMM. There is also a strong negative correlation between the expected value of $\|\phi_{distance}\|$ and the resulting accuracy, as expected from Ben-David *et al.*'s theoretical analysis. The HMM+E model outperforms the HMM on the (source) news domain by 0.3%, but actually performs worse for most target domains. We suspect that the entropy feature, which is learned only from labeled source-domain data, makes the representation biased towards features that are important in the source domain only. However, after we add in the distance bias and a parameter to balance the weights from both biases, the representation is able to capture the label information as well as the target domain features. Thus, the representation won't solely depend on source data. HMM+D+E, which combines both biases, outperforms HMM+D, suggesting that task-specific features for domain adaptation can be helpful, but only if there is some control for the domain-independence of the features.

5.2 English Named Entity Recognition

To evaluate on a second task, we turn to Named Entity Recognition. We use the training data from the

	news (source)	lore	reli	humour	gen-fic	essay	mystery	romance	sci-fi	skill	science	adv-fic	report	avg
words	9774	5428	3248	3326	4913	5214	5774	5489	3070	5464	5262	5071	6662	5284
CRF w/o HMM	93.8	85.0	80.0	85.4	85.0	83.8	84.7	86.0	82.8	78.2	82.2	77.1	85.3	84.5
HMM+E	97.1	88.2	83.1	87.5	87.4	89.2	89.5	87.1	86.7	82.1	87.2	79.4	91.7	88.3
Stanford	**98.8**	88.4	83.5	89.0	87.5	88.4	87.4	87.5	88.6	82.7	86.0	82.1	91.7	88.7
HMM	96.9	89.7	85.2	89.6	89.4	89.0	90.1	89.0	87.0	84.9	87.8	80.0	91.4	89.2
HMM+D	97.4	89.9	85.4	89.4	89.6	89.9	90.1	88.6	87.9	85.3	87.9	80.0	92.0	89.5
HMM+D+E	97.7	90.1	86.1	**89.8**	90.9	89.7	**90.3**	89.8	88.4	85.6	**87.9**	81.2	92.0	89.9
I-HMM+D+E	97.8	90.5	87.0	89.1	91.1	90.2	90.0	90.5	89.8	86.0	87.1	82.2	92.1	90.2
P-HMM+D+E	98.2	**91.5**	**87.7**	89.0	**91.8**	**91.0**	89.9	**91.4**	**90.4**	**87.0**	87.7	**83.4**	**92.4**	**90.8**

Table 2: **POS tagging accuracy: The P-HMM+D+E tagger outperforms the unbiased HMM tagger and the Stanford tagger on all target domains.** The 'avg' column includes source-domain development data results. Differences between the P-HMM+D+E and the Stanford tagger are statistically significant at $p < 0.01$ on average and on 11 out of 12 target domain. We used the two-tailed Chi-square test with Yates' correction.

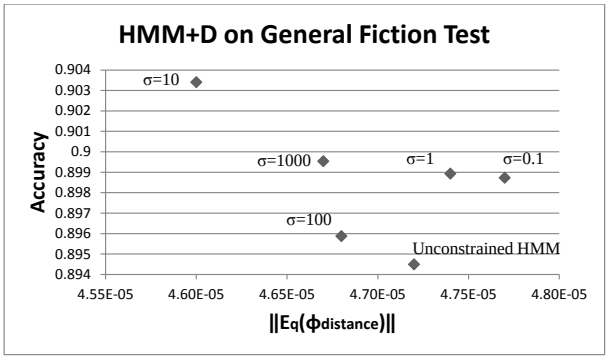

Figure 4: Greater distance between domains correlates with worse target-domain tagging accuracy.

System	F1
CRF without HMM	66.15
HMM+E	74.25
HMM	75.06
HMM+D	75.75
HMM+D+E	76.03
I-HMM+D+E	77.04
P-HMM+D+E	78.62

Table 3: English Named Entity recognition results

CoNLL 2003 shared task for our labeled training set, consisting of 204k tokens from the newswire domain. We tested the system on the MUC7 formal run test data, consisting of 59k tokens of stories on the telecommunications and aerospace industries.

To train our representations, we use the CoNLL training data and the MUC7 training data without labels. We again use a CRF, with features introduced by Zhang and Johnson (2003) for our baseline. We use the same setting of free parameters from our POS tagging experiments.

Results are shown in Table 3. Our best biased representation P-HMM+D+E outperformed the unbiased HMM representation by 3.6%, and beats the I-HMM+D+E by 1.6%. The domain-distance and multi-dimensional biases help most, while the task-specific bias helps somewhat, but only when the domain-distance bias is included. The best system tested on this dataset achieved a slightly better F1 score (78.84) (Turian et al., 2010), but used a much larger training corpus (they use RCV1 corpus which contains approximately 63 million tokens). Other studies (Turian et al., 2010; Huang et al., 2011) have performed a detailed comparison between these types of systems, so we concentrate on comparisons between biased and unbiased representations here.

5.3 Does the task-specific bias actually help?

In this section, we test whether the task-specific bias (entropy bias) actually learns something task-specific. We learn the entropy-biased representations for two tasks on the same set of sentences, labeled differently for the two tasks: English POS tagging and Named Entity Recognition. Then we switch the representations to see whether they will help or hurt the performance on the other task. We randomly picked 500 sentences from WSJ section

Representation/Task	POS Accuracy	NER F1
HMM	88.5	66.3
HMM+E(POS labels)	89.7	64.5
HMM+E(NER labels)	86.5	68.0

Table 4: Results of POS tagging and Named Entity recognition tasks with different representations. With the entropy-biased representation, the system has better performance on the task which the bias is trained for, but worse performance on the other task.

0-18 as our labeled training data and 500 sentences from WSJ section 20-23 as testing data. Because WSJ data does not have gold standard NER tags, we manually labeled these sentences with NER tags. For simplicity, we only use three types of NER tags: person, organization and location. The result is shown in Table 4. When the entropy bias uses labels from the same task as the classifier, the performance is improved: about 1.2% in accuracy on POS tagging and 1.7% in F1 score on NER. Switching the representations for the tasks actually hurts the performance compared with the unbiased representation. The results suggest that the entropy bias does indeed yield a task-specific representation.

6 Conclusion and Future Work

We introduce three types of biases into representation learning for sequence labeling using the PR framework. Our experiments on POS tagging and NER indicate domain-independent biases and multi-dimensional biases significantly improve the representations, while the task-specific bias improves performance on out-of-domain data if it is combined with the domain-independent bias. Our results indicate the power of representation learning in building domain-agnostic classifiers, but also the complexity of the task and the limitations of current techniques, as even the best models still fall significantly short of in-domain performance. Important considerations for future work include identifying further effective and tractable biases, and extending beyond sequence-labeling to other types of NLP tasks.

Acknowledgments

This research was supported in part by NSF grant IIS-1065397.

References

Arun Ahuja and Doug Downey. 2010. Improved extraction assessment through better language models. In *Proceedings of the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL-HLT)*.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA. MIT Press.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79:151–175.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. 2007. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*.

M. Candito and B. Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *IWPT*, pages 138–141.

M. Chang, L. Ratinov, and D. Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the ACL*.

Hal Daumé III, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the ACL Workshop on Domain Adaptation (DANLP)*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *ACL*.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via cca. In *Neural Information Processing Systems (NIPS)*.

A. Emami, P. Xu, and F. Jelinek. 2003. Using a connectionist model in a syntactical based language model. In *Proceedings of the International Conference on Spoken Language Processing*, pages 372–375.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:10–49.

Zoubin Ghahramani and Michael I. Jordan. 1997. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273.

Daniel Gildea. 2001. Corpus Variation and Parser Performance. In *Conference on Empirical Methods in Natural Language Processing*.

T. Honkela. 1997. Self-organizing maps of words for natural language processing applications. In *In Proceedings of the International ICSC Symposium on Soft Computing*.

Fei Huang and Alexander Yates. 2009. Distributional representations for handling sparsity in supervised sequence labeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Fei Huang and Alexander Yates. 2010. Exploring representation-learning approaches to domain adaptation. In *Proceedings of the ACL 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP)*.

Fei Huang, Alexander Yates, Arun Ahuja, and Doug Downey. 2011. Language models as representations for weakly supervised nlp tasks. In *Conference on Natural Language Learning (CoNLL)*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.

P. Liang, M. I. Jordan, and D. Klein. 2009. Learning from measurements in exponential families. In *International Conference on Machine Learning (ICML)*.

D. Lin and X Wu. 2009. Phrase clustering for discriminative learning. In *ACL-IJCNLP*, pages 1030–1038.

G. S. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *In Proc. ICML*.

Y. Mansour, M. Mohri, and A. Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*.

F. Morin and Y. Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, pages 246–252.

Sameer Pradhan, Wayne Ward, and James H. Martin. 2007. Towards robust semantic role labeling. In *Proceedings of NAACL-HLT*, pages 556–563.

M. Sahlgren. 2005. An introduction to random indexing. In *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE)*.

G. Salton and M.J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

Satoshi Sekine. 1997. The domain dependence of parsing. In *Proc. Applied Natural Language Processing (ANLP)*, pages 96–102.

Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 384–394.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Lijie Wang, Wanxiang Che, and Ting Liu. 2009. An svmtool-based chinese pos tagger. In *Journal of Chinese Information Processing*.

X. Yang, H. Fu, H. Zha, and J. Barlow. 2006. Semi-supervised nonlinear dimensionality reduction. In *Proceedings of the 23rd International Conference on Machine Learning*.

T. Zhang and D. Johnson. 2003. A robust risk minimization based named entity recognition system. In *CoNLL*.

D. Zhang, Z.H. Zhou, and S. Chen. 2007. Semi-supervised dimensionality reduction. In *Proceedings of the 7th SIAM International Conference on Data Mining*.