Exploring Representation-Learning Approaches to Domain Adaptation

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

Abstract

Most supervised language processing systems show a significant drop-off in performance when they are tested on text that comes from a domain significantly different from the domain of the training data. Sequence labeling systems like part-of-speech taggers are typically trained on newswire text, and in tests their error rate on, for example, biomedical data can triple, or worse. We investigate techniques for building open-domain sequence labeling systems that approach the ideal of a system whose accuracy is high and constant across domains. In particular, we investigate unsupervised techniques for representation learning that provide new features which are stable across domains, in that they are predictive in both the training and out-of-domain test data. In experiments, our novel techniques reduce error by as much as 29% relative to the previous state of the art on out-of-domain text.

1 Introduction

Supervised natural language processing (NLP) systems exhibit a significant drop-off in performance when tested on domains that differ from their training domains. Past research in a variety of NLP tasks, like parsing (Gildea, 2001) and chunking (Huang and Yates, 2009), has shown that systems suffer from a drop-off in performance on out-of-domain tests. Two separate experiments with part-of-speech (POS) taggers trained on Wall Street Journal (WSJ) text show that they can reach accuracies of 97-98% on WSJ test sets, but achieve accuracies of at most 90% on biomedical text (R.Codena et al., 2005; Blitzer et al., 2006).

The major cause for poor performance on out-of-domain texts is the traditional representation used by supervised NLP systems. Most systems depend to varying degrees on lexical features, which tie predictions to the words observed in each example. While such features have been used in a variety of tasks for better in-domain performance, they are pitfalls for out-of-domain tests for two reasons: first, the vocabulary can differ greatly between domains, so that important words in the test data may never be seen in the training data. And second, 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.

Representation learning is a promising new approach to discovering useful features that are stable across domains. Blitzer *et al.* (2006) and our previous work (2009) demonstrate novel, unsupervised representation learning techniques that produce new features for domain adaptation of a POS tagger. This framework is attractive for several reasons: experimentally, learned features can yield significant improvements over standard supervised models on out-of-domain tests. Since the representation learning techniques are unsupervised, they can be applied to arbitrary new domains to yield the best set of features for learning on WSJ text and predicting on the new domain. There is no need to supply additional labeled examples for each new domain. This reduces the effort for domain adaptation, and makes it possible to apply systems to open-domain text collections like the Web, where it is prohibitively expensive to collect a labeled sample that is truly representative of all domains.

Here we explore two novel directions in the representation-learning framework for domain adaptation. Specifically, we investigate empirically the effects of representation learning techniques on POS tagging to answer the following:

Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing, ACL 2010, pages 23–30,
Uppsala, Sweden, 15 July 2010.

1. Can we produce multi-dimensional representations for domain adaptation? Our previous efforts have provided only a single new feature in the learned representations. We now show how we can perform a multi-dimensional clustering of words such that each dimension of the clustering forms a new feature in our representation; such multi-dimensional representations dramatically reduce the out-of-domain error rate of our POS tagger from 9.5% to 6.7%.
2. Can maximum-entropy models be used to produce representations for domain adaptation? Recent work on contrastive estimation (Smith and Eisner, 2005) has shown that maximum-entropy-based latent variable models can yield more accurate clusterings for POS tagging than more traditional generative models trained with Expectation-Maximization. Our preliminary results show that such models can be used effectively as representations for domain adaptation as well, matching state-of-the-art results while using far less data.

The next section provides background information on learning representations for NLP tasks using latent-variable language models. Section 3 describes our experimental setup. In Sections 4 and 5, we empirically investigate our two questions with a series of representation-learning methods. Section 6 analyzes our best learned representation to help explain its effectiveness. Section 7 presents previous work, and Section 8 concludes and outlines directions for future work.

2 Open-Domain Sequence Labeling by Learning Representations

Let $\mathcal{X}$ be an instance set for a learning problem; for POS tagging, for instance, this could be the set of all English sentences. Let $\mathcal{Y}$ be the space of possible labels for an instance, and let f: $\mathcal{X} \rightarrow \mathcal{Z}$ be the target function to be learned. A *representation* is a function R: $\mathcal{X} \rightarrow \mathcal{Y}$, for some suitable feature space $\mathcal{Y}$ (such as $\mathbb{R}^{\mathrm{d}}$). A *domain* is defined as a distribution $\mathcal{D}$ over the instance set $\mathcal{X}$. An open-domain system observes a set of training examples $(R(x), f(x))$, where instances $x \in \mathcal{X}$ are drawn from a *source* domain, to learn a hypothesis for classifying examples drawn from a separate *target* domain.

Previous work by Ben-David *et al.* (2007) uses Vapnik-Chervonenkis (VC) theory to show that the choice of representation is crucial to open-domain learning. As is customary in VC theory, a good choice of representation must allow a learning machine to achieve low error rates during training. Just as important, however, is that *the representation must simultaneously make the source and target domains look as similar to one another as possible*.

For open-domain sequence-labeling, then, the traditional representations are problematic. Typical representations in NLP use functions of the local context to produce features. Although many previous studies have shown that such lexical features allow learning systems to achieve impressively low error rates during training, they also make texts from different domains look very dissimilar. For instance, a sentence containing "bank" is almost certainly from the WSJ rather than biomedical text; a sentence containing "pathway" is almost certainly from a biomedical text rather than from the WSJ.

Our recent work (2009) shows how to build systems that learn new representations for open-domain NLP using latent-variable language models like Hidden Markov Models (HMMs). In POS-tagging and chunking experiments, these learned representations have proven to meet both of Ben-David *et al.*'s criteria for representations. They help discriminate among classes of words, since HMMs learn distributional similarity classes of words that often correlate with the labels that need to be predicted. Moreover, it would be difficult to tell apart two domains based on the set of HMM states that generated the texts, since a given HMM state may generate words from any number of domains.

In the rest of this paper, we investigate ways to improve the predictive power of the learned representations, without losing the essential property that the features remain stable across domains. We stay within the framework of using graphical models to learn representations, and demonstrate significant improvements on our original technique.

3 Experimental Setup

We use the same experimental setup as Blitzer *et al.* (2006): the Penn Treebank (Marcus et al., 1993) Wall Street Journal portion for our labeled training data; 561 MEDLINE sentences (9576 words) from the Penn BioIE project (PennBioIE, 2005) for our labeled test set; and all of the unlabeled text from the Penn Treebank WSJ portion plus Blitzer *et al.*'s MEDLINE corpus of 71,306

unlabeled sentences to train our latent variable models. The two texts come from two very different domains, making this data a tough test for domain adaptation. 23% of the word types in the test text are Out-Of-Vocabulary (OOV), meaning that they are never observed in the training data.

We use a number of unsupervised representation learning techniques to discover features from our unlabeled data, and a supervised classifier to train on the training set annotated with learned features. We use an open source Conditional Random Field (CRF) (Lafferty et al., 2001) software package[1] designed by Sunita Sajarwal and William W. Cohen to implement our supervised models. We refer to the baseline system with feature set following our previous work (2009) as PLAIN-CRF. Our learned features will supplement this set.

For comparison, we also report on the performance of Blitzer *et al.*'s Structural Correspondence Learning (SCL) (2006), our HMM-based model (2009)(HY09), and two other baselines:

- TEST-CRF: Our baseline model, trained and tested on the test data. This is our upper bound.
- SELF-CRF: Following the self-training paradigm (*e.g.*, (McClosky et al., 2006b; McClosky et al., 2006a)), we train our baseline first on the training set, then apply it to the test set, then retrain it on the training set plus the automatically labeled test set. We perform only one iteration of retraining, although in general multiple iterations are possible, usually with diminishing marginal returns.

4 Multi-dimensional Representations

From a linguistic perspective, words are multi-dimensional objects. For instance, the word "we" in "We like doing domain adaptation research" is a pronoun, a subject, first person, and plural, among other things. Each of these properties is a separate feature of this word, which can be changed without changing the other features. For example, if "we" is changed to "they" in the previous example, it is exactly the same as "we" in all aspects, except that it is third person; if "we" is changed to "us", then it changes from subject case to object case. In morphologically rich languages, many syntactic distinctions are marked in the surface forms of words; in more analytic or isolating languages like English, the distinctions are still there, but must often be inferred from context rather than word form. Beyond syntactic dimensions, numerous semantic properties can also distinguish words, such as nouns that refer to cognitive agents versus nouns that refer to materials and tools.

We seek to learn multidimensional representations of words. Our HMM-based model is able to categorize words in one dimension, by assigning a single HMM latent state to each word. Since the HMM is trained on unlabeled data, this dimension may partially reflect POS categories, but more likely represents a mixture of many different word dimensions. By adding in multiple hidden layers to our sequence model, we aim to learn a multi-dimensional representation that may help us to capture word features from multiple perspectives. The supervised CRF system can then sort out which dimensions are relevant to the sequence-labeling task at hand.

A Factorial HMM (FHMM) can be used to model multiple hidden dimensions of a word. However, the memory requirements of an FHMM increase exponentially with the number of layers in the graphical model, making it hard to use (see Table 1). Although other parameterizations may require much less memory, like using a log-linear output distribution conditioned on the factors, exact inference is still computationally intractable; exploring FHMMs with approximate inference and learning is an interesting area for future work. Here, we choose to create several single-layer HMMs separately. Figure 1 shows our Independent-HMM model (I-HMM). I-HMM has several copies of the observation sequence and each copy is associated with its own hidden label sequence. To encourage each layer of the I-HMM model to find a different local maximum in parameter space during training (and thus a different model of the observation sequence), we initialize the parameters randomly.

Suppose there are L independent layers in an I-HMM model for corpus $\mathbf{x} = (x_1, \ldots, x_N)$, and each layer is ($y_1^l$,$y_2^l$,...$y_N^l$), where $l = 1...L$ and each y can have K states. The distribution of the corpus and one hidden layer l is

$$P(\mathbf{x}, \mathbf{y^l}) = \prod_i P(x_i|y_i^l)P(y_i^l|y_{i-1}^l)$$

For each layer l, for each position i, each HMM

[1] Available from http://sourceforge.net/projects/crf/

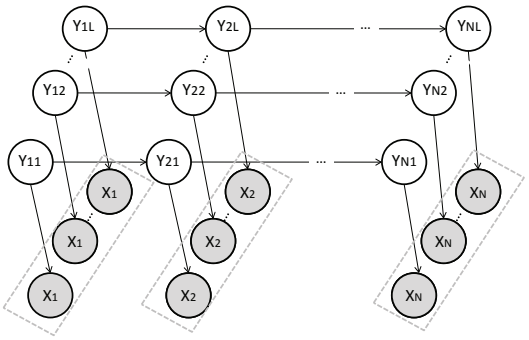

Figure 1: Graphical models of an Independent Hidden Markov Model. The dash line rectangle indicates that they are copies of the observation sequence

Model	Number of layers	words	states	Memory
HMM	1	W	K	$O(WK + K^2)$
FHMM	L	W	K	$O(WK^L + LK^2)$
I-HMM	L	W	K	$O(WKL + LK^2)$

Table 1: The memory requirement for HMM, FHMM, and I-HMM models.

state y and each POS tag z, we add a new boolean feature to our CRF system that indicates whether $Y_i^l = y$ and Z_i=z.

We experiment with two versions of I-HMM: first, we fix the number of states in each layer at 80 states, and increase the number of HMM layers from 1 to 8 (I-HMM(80)). Second, to provide greater encouragement for each layer to represent separate information, we vary the number of states in each layer (I-HMM(vary)). The detailed configuration for this model is shown in Table 2.

The results for our two models are shown in Figure 2. We can see that the accuracy of I-HMM(80) model keeps increasing from 90.5% to 93.3% until 7 layers of HMM features (we call this 7-layer representation I-HMM*). This is a dramatic 29% decrease in the best reported error rate for this dataset when no labeled data from the biomedical domain is used. Unlike with an FHMM, there is no guarantee that the different layers of an I-HMM will model different aspects of the observation signal, but our results indicate that for at least several layers, the induced models are complementary. After 7 layers, results begin to decrease, most likely because the added layer is no longer complementary to the existing latent-variable models and is causing the supervised CRF to overfit the training data.

For the I-HMM(vary) model with up to 5 layers, the accuracy is not as good as I-HMM(80), although the 5-layer model still outperforms HY09. Individually, HMM models with fewer than 80 states perform worse than the 80-state model (a model with 40 states achieved 89.4% accuracy, and a model with 20 states achieved 88.9%). We had hoped that by using layers with different numbers of states, we could force the layers to learn complementary models, but the results indicate that any benefit from complementarity is outweighed by the lower performance of the individual layers.

Number of Layers	Number of States in each Layer
1	10
2	10 20
3	10 20 40
4	10 20 40 60
5	10 20 40 60 80

Table 2: The configuration of HMM layers and HMM states for the I-HMM(vary) model

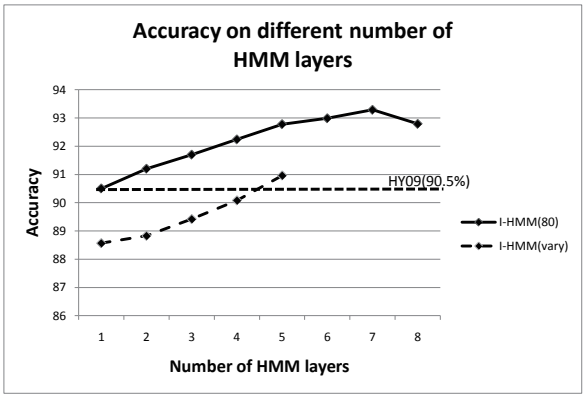

Figure 2: Our best multi-dimensional smoothed-HMM tagger with 7 layers reaches 93.3% accuracy, a drop of nearly 3% in the error rate from the previous state of the art (HY09).

5 Learning Representations with Contrastive Estimation

In recent years, many NLP practitioners have begun using discriminative models, and especially maximum-entropy-based models like CRFs, because they allow the modeler to incorporate arbitrary, interacting features of the observation sequence while still providing tractable inference. To see if the same benefit can carry over to our representation learning, we aim to build maximum-entropy-based linear-chain models that, unlike

most discriminative models, train on unannotated data. We follow Smith and Eisner (2005) in training our models using a technique called *contrastive estimation*, which we explain below. We call the resulting model the Smith and Eisner Model (SEM).

The key to SEM is that the contrastive estimation training procedure forces the model to explain why the given training data are better than perturbed versions of the data, called neighbor points. For example, the sentence "We like doing domain adaptation research" is a valid sentence, but if we switched "like" and "doing", the new sentence "We doing like domain adaptation research" is not valid. SEM learns a model of the original sentence by contrasting it with the invalid neighbor sentences.

Let $\vec{x} =< x_1, x_2, ..., x_N >$ be the observed example sentences, and let $\mathcal{Y}$ be the space of possible hidden structures for x_i. Let $\mathcal{N}(x_i)$ be a "neighborhood" for x_i, or a set of negative examples obtained by perturbing x_i, plus x_i itself. Given a vector of feature functions $\vec{f}(x, y)$, SEM tries to find a set of weights $\vec{\theta}$ that maximize a log-likelihood function:

$$\mathcal{L}_{\mathcal{N}}(\vec{\theta}) = log \prod_i \frac{\sum_{y \in \mathcal{Y}} u(x_i, y|\vec{\theta})}{\sum_{(x,y) \in \mathcal{N}(x_i) \times \mathcal{Y}} u(x, y|\vec{\theta})}$$

where $u(x, y|\vec{\theta}) = \exp(\vec{\theta} \cdot \vec{f}(x, y))$ is the "unnormalized probability" of an (example, hidden structure) pair (x,y). Following Smith and Eisner, we use the best performing neighborhood, called TRANS1, to conduct our experiments. TRANS1 is the set of sentences resulting from transposing any pair of adjacent words for any given training example.

The base feature space for SEM includes two kinds of boolean features analogous to HMM emission and transition probabilities. For an observation sequence $x_1, \ldots, x_T$ and a label sequence $y_1, \ldots, y_T$, a boolean emission feature indicates whether $x_t = x$ and $y_t = y$ for all possible t, x, and y. A boolean transition feature indicates whether $y_{t-1} = y$ and $y_t = y'$ for all possible t, y, and y'.

Because contrastive estimation is a computationally expensive training procedure, we take two steps to reduce the computational cost: we reduce the unlabeled data set, and we prune the feature set of SEM. For our training data, we use only the sentences with length less than or equal to 10. We also get rid of punctuation and the corresponding tags, change all words to lowercase and change all numbers into a single symbol.

To reduce the feature space, we create a tagging dictionary from Penn Treebank sections 02-21: for every word in these sections, the dictionary records the set of POS tags that were ever associated with that word. We then prune the emission features for words that appear in this dictionary to include only the features that associate words with their corresponding POS tags in the dictionary. For the words that don't appear in the Penn Treebank, they are associated with all possible POS tags. This procedure reduces the total number of features in our SEM model from over 500,000 to just over 60,000.

After we train the model, we use a Viterbi-like algorithm to decode it on the testing set. Unlike the HMM model, the decoded states of SEM are already meaningful POS tags, so we can use these decoded states as POS tags (PLAIN-SEM), or use them as features for a CRF model (SEM-CRF). We show the result of both models, as well as several comparison models, in Table 3. From the result, we can see that the unsupervised PLAIN-SEM outperforms the supervised PLAIN-CRF on both all words and OOV words. This impressive performance results from its ability to adapt to the new domain through the unlabeled training examples and the contrastive estimation training procedure. In addition, the SEM-CRF model significantly outperforms the SCL model (88.9%) and the HMM-based CRF with 40 hidden states (89.4%) while using only 36 hidden states, although it does not quite reach the performance of HY09. These results, which use a subset of the available unlabeled training text, suggest that maximum-entropy-style representation learning is a promising area for further investigation.

6 Analysis

As we mention in Section 2, the choice of representation is crucial to open-domain learning. In Sections 4 and 5, we demonstrate empirically that learned representations based on latent-variable graphical models can significantly improve the accuracy of a POS tagger on a new domain, compared with using the traditional word-level representations. We now examine our best representation, I-HMM*, in light of the theoretical predictions made by VC theory.

Model	All words	OOV words
PLAIN-CRF	88.3	67.3
SELF-CRF	88.5	70.4
PLAIN-SEM	88.5	69.8
SCL	88.9	72.0
SEM-CRF	90.0	71.9
HY09	90.5	75.2
I-HMM*	**93.3**	**76.3**
TEST-CRF	98.9	NA

Table 3: SEM-CRF reduces error compared with SCL by 1.1% on all words; I-HMM* closes 33% of the gap between the state-of-the-art HY09 and the upper-bound, TEST-CRF.

In particular, Ben-David *et al.*'s analysis shows that the distance between two domains under a representation R of the data is crucial to domain adaptation. However, their analysis depends on a particular notion of distance, the $\mathcal{H}$-divergence, that is computationally intractable to calculate. For our analysis, we resort instead to a crude but telling approximation of this measure, using a more standard notion of distance: Jensen-Shannon Divergence (D_{JS}).

To calculate the distance between domains under a representation R, we represent a domain D as a multinomial probability distribution over the set of features in R. We take maximum-likelihood estimates of this distribution using our samples from the WSJ and MEDLINE domains. We then measure the Jensen-Shannon Divergence between the two distributions, which for discrete distributions is calculated as

$$D_{JS}(p||q) = \frac{1}{2}\sum_i \left[p_i \log\left(\frac{p_i}{m_i}\right) + q_i \log\left(\frac{q_i}{m_i}\right)\right]$$

where $m = \frac{p+q}{2}$.

Figure 3 shows the divergence between these two domains under purely lexical features, and under only HMM-based features. OOV words make up a substantial portion of the divergence between the two domains under the lexical representation, but even if we ignore them the HMM features are substantially less variable across the two domains, which helps to explain their ability to provide supervised classifiers with stable features for domain adaptation. Because there are so few HMM states compared with the number of word types, there is no such thing as an OOV HMM state, and the word

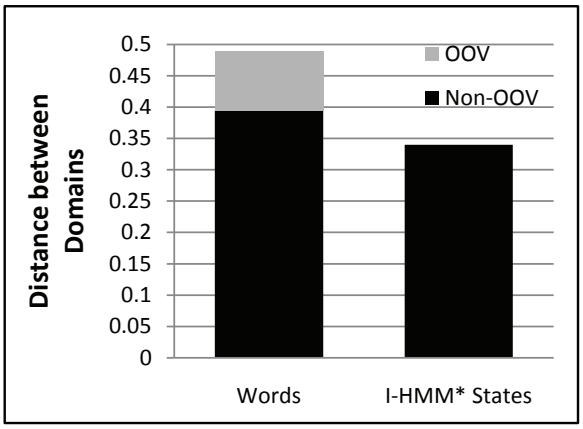

Figure 3: The Jensen-Shannon Divergence between the newswire domain and the biomedical domain, according to a word-based representation of the domains and a HMM-based representation. The portion of the distance that is due to words which appear in the biomedical domain but not the newswire domain is shown in gray.

states that appear in training data appear roughly as often in test data. This means that any associations that the CRF might learn between HMM states and predicted outcomes is likely to remain useful on the test data, but associations between words and outcomes are less likely to be useful.

7 Previous Work

Previous work on artificial neural networks (ANNs) (Fahlman and Lebiere, 1990) has shown that it is possible to learn effectively by adding more hidden units to the neural network that correlate with the residual error of the existing hidden units (Cascade-Correlation learning). Like our I-HMM technique, this work aims to build a multi-dimensional model, and it is capable of learning the number of appropriate dimensions. Unlike the ANN scenario, our multi-dimensional learning techniques must handle unlabeled data, and they rely on the sequential structure of language to learn effectively, whereas Cascade-Correlation learning assumes samples are independent and identically distributed. Our techniques do not (yet) automatically determine the best number of layers in the model.

Unlike our techniques for domain adaptation, in most cases researchers have focused on the scenario where labeled training data is available in both the source and the target domain (*e.g.*, (Bacchiani et al., 2006; Daumé III, 2007; Chelba and

Acero, 2004; Daumé III and Marcu, 2006; Blitzer et al., 2007)). Our techniques use only raw text from the target domain. This reduces the cost of domain adaptation and makes the techniques more widely applicable to new domains like web processing, where the domain and vocabulary is highly variable, and it is extremely difficult to obtain labeled data that is representative of the test distribution. When labeled target-domain data is available, instance weighting and similar techniques can potentially be used in combination with our techniques to improve our results further.

Several researchers have previously studied methods for using unlabeled data for sequence labeling, either alone or as a supplement to labeled data. Ando and Zhang develop a semi-supervised chunker that outperforms purely supervised approaches on the CoNLL 2000 dataset (Ando and Zhang, 2005). Recent projects in semi-supervised (Toutanova and Johnson, 2007) and unsupervised (Biemann et al., 2007; Smith and Eisner, 2005) tagging also show significant progress. HMMs have been used many times for POS tagging in supervised, semi-supervised, and in unsupervised settings (Banko and Moore, 2004; Goldwater and Griffiths, 2007; Johnson, 2007). The REALM system for sparse information extraction has also used unsupervised HMMs to help determine whether the arguments of a candidate relation are of the appropriate type (Downey et al., 2007). Schütze (1994) has presented an algorithm that categorizes word tokens in context instead of word types for tagging words. We take a novel perspective on the use of unsupervised latent-variable models by using them to compute features of each token that represent the distribution over that token's contexts. These features prove to be highly useful for supervised sequence labelers in out-of-domain tests.

In the deep learning (Bengio, 2009) paradigm, researchers have investigated multi-layer latent-variable models for language modeling, among other tasks. While n-gram models have traditionally dominated in language modeling, two recent efforts develop latent-variable probabilistic models that rival and even surpass n-gram models in accuracy (Blitzer et al., 2005; Mnih and Hinton, 2007). Several authors investigate neural network models that learn a vector of latent variables to represent each word (Bengio et al., 2003; Emami et al., 2003; Morin and Bengio, 2005). And factorial Hidden Markov Models (Ghahramani and Jordan, 1997) are a multi-layer variant of the HMM that has been used in speech recognition, among other things. We use simpler mixtures of single-layer models for the sake of memory-efficiency, and we use our models as representations in a supervised task, rather than as language models.

8 Conclusion and Future Work

Our representation learning approach to domain adaptation yields state-of-the-art results in POS tagging experiments. Our best models use multi-dimensional clustering to find several latent categories for each word; the latent categories serve as useful and domain-independent features for our supervised learner. Our exploration has yielded significant progress already, but it has only scratched the surface of possible models for this task. The current representation learning techniques we use are unsupervised, meaning that they provide the same set of categories, regardless of what task they are to be used for. Semi-supervised learning approaches could be developed to guide the representation learning process towards features that are best-suited for a particular task, but are still useful across domains. Our current approach also requires retraining of a CRF for every new domain; incremental retraining techniques for new domains would speed up the process and make domain adaptation much more accessible. Finally, there are cases where small amounts of labeled data are available for new domains; models that combine our representation learning approach with instance weighting and other forms of supervised domain adaptation may take better advantage of these cases.

Acknowledgments

We wish to thank the anonymous reviewers for their helpful comments and suggestions.

References

Rie Kubota Ando and Tong Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *ACL*.

Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Michele Banko and Robert C. Moore. 2004. Part of speech tagging in context. In *COLING*.

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.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Y. Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2.

C. Biemann, C. Giuliano, and A. Gliozzo. 2007. Unsupervised pos tagging supporting supervised methods. *Proceeding of RANLP-07*.

J. Blitzer, A. Globerson, and F. Pereira. 2005. Distributed latent variable models of lexical cooccurrences. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*.

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*.

Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy classifier: Little data can help a lot. In *EMNLP*.

Hal Daumé III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26.

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

Doug Downey, Stefan Schoenmackers, and Oren Etzioni. 2007. Sparse information extraction: Unsupervised language models to the rescue. In *ACL*.

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.

Scott E. Fahlman and Christian Lebiere. 1990. The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems 2*.

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*.

Sharon Goldwater and Thomas L. Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *ACL*.

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*.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers. In *EMNLP*.

J. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proc. of HLT-NAACL*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 337–344.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648, New York, NY, USA. ACM.

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.

PennBioIE. 2005. Mining the bibliome project. *http://bioie.ldc.upenn.edu/*.

Anni R.Codena, Serguei V.Pakhomovb, Rie K.Andoa, Patrick H.Duffyb, and Christopher G.Chute. 2005. Domain-specific language models and lexicons for tagging. *Journal of Biomedical Informatics*, 38(6):422–430.

Hinrich Schütze. 1994. Distributional part-of-speech tagging. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 354–362, Ann Arbor, Michigan, June.

Kristina Toutanova and Mark Johnson. 2007. A bayesian LDA-based model for semi-supervised part-of-speech tagging. In *NIPS*.