Semi-supervised Representation Learning for Domain Adaptation using Dynamic Dependency Networks

Min Xiao *Yuhong Guo* *Alexander Yates*
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA
`{minxiao, yuhong, yates}@temple.edu`

ABSTRACT
Recently, various unsupervised representation learning approaches have been investigated to produce augmenting features for natural language processing systems in the open-domain learning scenarios. In this paper, we propose a dynamic dependency network model to conduct semi-supervised representation learning. It exploits existing task-specific labels in the source domain in addition to the large amount of unlabeled data from both the source and target domains to produce informative features for NLP tasks. We empirically evaluate the proposed learning technique on the part-of-speech tagging task using Wall Street Journal and MEDLINE sentences and on the syntactic chunking task using Wall Street Journal corpus and Brown corpus. Our experimental results show that the proposed semi-supervised learning model can produce more effective features than unsupervised representation learning methods for open-domain part-of-speech taggers and syntactic chunkers.

KEYWORDS: Domain Adaptation, Representation Learning, POS Tagging, Syntactic Chunking.

1 Introduction

Existing supervised natural language processing (NLP) systems are highly domain-dependent, whose performance degrades significantly when tested on a new domain. Previous works in a variety of NLP tasks, like part-of-speech (POS) tagging (Blitzer et al., 2006; Huang and Yates, 2010; Blitzer et al., 2011), syntactic chunking (Huang and Yates, 2009; Carreras and Màrquez, 2005), named entity recognition (NER) (Daumé III, 2007; Turian et al., 2010; Daumé III and Marcu, 2006), or parsing (Sekine, 1997; McClosky et al., 2010) show that the performance of supervised NLP systems drops a lot on domains whose vocabulary differs from the vocabulary of the training data.

The major reason that causes the increasing of test error on out-of-domain texts is the traditional representation used in the supervised NLP systems. Most NLP systems use the lexical features for predictions. Though it works very well for various in-domain NLP tasks, they perform poorly when tested on a different domain. There are two main reasons. First, the source and target domains may have very different vocabularies, thus some test words may never appear during the training phase. For example, “sequencing”, “metastases” and “genomic” show up frequently as lexical features in biomedical text but rarely in newswire articles. A classifier trained on newswire data thus will have seen few training examples related to sentences with lexical features “sequencing”, “metastases” and “genomic” (Ben-David et al., 2010, 2007). Second, the prediction function based on lexical features may change across domains. For example, “signaling” appears in “signaling that ...” from a Wall Street Journal (WSJ) article primarily as a present participle (VBG) (Marcus et al., 1993), but predominantly as a noun in “signaling pathway” from a MEDLINE text (PennBioIE, 2005).

Recently, various unsupervised representation learning techniques are proposed to induce generalizable latent features across domains by exploiting large amount of unlabeled data from both the source and the target domains. Blitzer et al. (2006) and Huang and Yates (2009, 2010) show that their learned representations can yield significant improvements for out-of-domain POS taggers or syntactic chunkers. However, the latent features produced by these unsupervised representation learning techniques provide no task-specific discriminative information over the labels of NLP tasks.

To tackle this issue, in this paper we propose a semi-supervised Dynamic Dependency Network (DDN) model to induce task-specific discriminative latent features across domains. In addition to exploiting large amount of unlabeled data from two domains, the DDN model will also leverage the already-existing task labels from the source domain. It combines the advantages of semi-supervised learning methods from (Blitzer et al., 2006; Daumé III, 2007) with the sequence models from (Huang and Yates, 2009, 2010), while maintaining desirable properties like computational tractability and modeling flexibility to incorporate many features. This model is more appealing than unsupervised representation learning techniques when a target NLP task is known. Moreover, though we perform representation learning in a semi-supervised manner, we only exploit the existing labeled data in the source domain. Thus our model can be applied to arbitrary new domains without any extra annotation effort. The proposed model is empirically evaluated for out-of-domain POS tagging systems on articles from WSJ and MEDLINE, and for out-of-domain syntactic chunking systems on articles from WSJ and Brown corpora. It is shown to outperform unsupervised representation learning techniques. Overall, the contributions of this paper include

- We propose a novel probabilistic graphical model, Dynamic Dependency Networks

(DDNs), which is computationally tractable for inference and training.

- We demonstrate how to apply DDNs on cross-domain semi-supervised representation learning for sequence labeling systems.

- Our empirical results show that DDN-based semi-supervised representation learning is superior to unsupervised representation learning for out-of-domain POS tagging and syntactic chunking.

The remainder of the paper is organized as follows. The next section discusses previous work. Section 3 describes representation learning. Section 4 presents the proposed DDN model and semi-supervised representation learning. Section 5 presents experimental results for out-of-domain POS tagging systems. Section 6 presents empirical results for out-of-domain syntactic chunking systems. We then conclude the paper.

2 Previous Work

Most previous work for domain adaptation tasks has focused on the setting where some labeled data is available in the target domain (Daumé III and Marcu, 2006; Daumé III, 2007; Jiang and Zhai, 2007; Dredze et al., 2010; Daumé III et al., 2010). Daumé III and Marcu (2006) proposed to tackle domain adaptation tasks by training three separate models to distinguish source-specific, target-specific and general information using maximum entropy classifiers. Jiang and Zhai (2007) adopted instance weighting method for semi-supervised domain adaptation by removing misleading training instances in the source domain, assigning more weights to labeled data, and augmenting training data using target instances with predicted labels. Daumé III (2007) proposed to perform supervised domain adaptation with feature augmentation for various NLP tasks. Daumé III et al. (2010) used co-regularization to incorporate unlabeled data for semi-supervised domain adaptation. In contrast, we investigate a more practical setting for domain adaptation where we have no labeled data in the target domain.

Recently, various unsupervised representation learning techniques have been proposed to tackle domain adaptation tasks by exploiting large amount of unlabeled data from two domains (Ando and Zhang, 2005; Blitzer et al., 2006; Huang and Yates, 2009, 2010; Blitzer et al., 2011). Blitzer et al. (2006) proposed a structural correspondence learning (SCL) method to seek for generalizable features by modeling the correlation between pivot features and non-pivot features. Turian et al. (2010) empirically evaluated Collobert and Weston embeddings (Collobert and Weston, 2008), Brown clusters, and HLBL embeddings (Mnih and Hinton, 2009) of words on both syntactic chunking and named entity recognition tasks. Their experimental results demonstrated that those three word representations can improve the performance of out-of-domain named entity recognition systems and in-domain syntactic chunking systems. Huang and Yates (2009) employed Hidden Markov Models (HMMs) to induce hidden states of the sentence words as latent features. Later, Huang and Yates (2010) proposed to learn a multi-dimensional feature representation by simultaneously train multiple HMMs with different initializations. Though unsupervised representation learning achieves good empirical performance for out-of-domain NLP tasks, it underutilizes the source data, since it completely neglects the existing task-specific labels when performing representation learning. The DDN model we propose in this work can suitably address this problem by exploiting task labels when performing semi-supervised representation learning.

3 Representation Learning

A *representation* is a set of features describing instances in a classification problem. Let $\mathscr{X}$ be the set of all instances. For example, for a sequence labeling task in NLP, $\mathscr{X}$ is the set of all sentences. Let $\mathscr{Z}$ be the label set of the classification problem. For POS tagging, $\mathscr{Z}$ is the set of all sequences of part-of-speech tags. For syntactic chunking, $\mathscr{Z}$ is the set of all sequences of syntactic chunks. Let $f : \mathscr{X} \rightarrow \mathscr{Z}$ be the prediction function. A representation is a function $R : \mathscr{X} \rightarrow \mathscr{Y}$, for some suitable feature space $\mathscr{Y}$ (such as $\mathbb{R}^d$). A *domain* $\mathscr{D}$ is defined as a distribution over the instance set $\mathscr{X}$. An open-domain system learns a classification model from a set of training instances $(R(x), f(x))$, where each instance $x \in \mathscr{X}$ is drawn from a *source* domain $\mathscr{D}_s$ and expressed in a representation space defined by function R, and classifies test instances drawn from a separate *target* domain $\mathscr{D}_t$.

It has been shown in recent theoretical work that the performance of domain adaptation greatly depends on the data representation employed, and traditional data representations in NLP prevent learning systems from generalizing appropriately across domains (Ben-David et al., 2010). Previous work by Ben-David et al. (2007) uses Vapnik-Chervonenkis (VC) theory (Vapnik, 1995) to prove theoretical bounds on an open-domain learning machine's performance. It demonstrates that the choice of representation is crucial for domain adaptation. It is customary in VC theory that a good choice of representation must allow a learning machine to achieve low error rates during training.

In light of Ben-David et al.'s theory findings, traditional representations in NLP are inadequate or problematic for domain adaptation. Traditional representations in NLP tasks are lexical features based on local context. Although many previous studies have shown that lexical features allow learning systems to achieve impressively low error rates during training, they also make texts from different domains look very dissimilar and create domain divergence problems. For example, a sentence containing "CEO" may be common in a domain of newswire text but scarce or nonexistent in a different domain like biomedical articles. Likewise, a sentence containing "path-way" is almost certainly from a biomedical literature rather than from a newswire article. Thus with traditional representations of NLP, a prediction model trained in one source domain can hardly work well in a different target domain.

At the same time, traditional representations contribute to *data sparsity*, a lack of sufficient training data for the relevant parameters of the system. In traditional supervised NLP systems, there are parameters for each word type in the data, or perhaps even combinations of word types. Since vocabularies can be extremely large, this leads to an explosion in the number of parameters. As a consequence, for many of their parameters, supervised NLP systems have zero or only a handful of labeled examples. No matter how sophisticated the learning technique, it is difficult to estimate parameters without relevant data. Because vocabularies differ across domains, domain adaptation greatly exacerbates this issue of data sparsity.

Huang and Yates (2009) show how to use language models, HMMs, to induce latent-variable states as generalizable features for various open-domain NLP tasks, such as POS tagging and syntactic chunking. These learned representations have proven to meet the criteria for open-domain representations. It would be difficult to tell two domains apart based on the HMM labels since the same HMM states may generate many similar words from a variety of domains. However, these unsupervised representations are not specifically discriminative for any NLP tasks. This is the main motivation of the research in this paper. Unsupervised representation learning based on HMMs nevertheless serves as one of the comparisons in our experiments.

4 Dynamic Dependency Network for Semi-supervised Representation Learning

In this section, we present a Dynamic Dependency Network (DDN) model to incorporate task-specific label information in the source domain for semi-supervised representation learning. A dynamic dependency network is a dynamic extension of dependency networks (Heckerman et al., 2000) for modeling data with sequential observations and labels. Dependency networks are cyclic directed graphical models. Similar to directed acyclic Bayesian networks, dependency networks allow simple local parameter estimations given fully observed data. But by dropping acyclicity constraints, dependency networks are more flexible on modeling interdependencies between variables than acyclic Bayesian networks. Following the same principle of Dynamic Bayesian Networks (DBNs) (Murphy, 2002), we extend dependency networks into sequential models to form Dynamic Dependency Networks. Although with directed cycles a DDN model will lose the ability of handling time series data that requires time forward directed arcs (not vice versa), it has increased the capacity of modeling word or label interdependencies within local contexts of sentences, comparing to DBNs. Figure 1 demonstrates an example of the DDN models we will use for semi-supervised representation learning.

In this DDN model (Figure 1), the variables are partitioned into three interconnected sequences $X = \{X_1, \ldots, X_T\}$, $Y = \{Y_1, \ldots, Y_T\}$ and $Z = \{Z_1, \ldots, Z_T\}$, representing observations, hidden states and labels respectively. Similar to the HMM model used in (Huang and Yates, 2009), the state sequence is hidden in our model and the state variable Y_t at location t takes values from a predefined set of state values; the observation sequence X is produced from the observed sentence; given Y_t, we assume X_t is conditionally independent of $X_{t'}$ for $t \neq t'$. But in addition to the two layers, X and Y in HMMs, our DDN model adds another task-specific label layer Z. For example, for the POS tagging task, Z will be the sequence of POS tags. Moreover, we take the bi-directional sequential dependency between labels into consideration by connecting each neighbor pairs of labels using bi-directional arcs. At each location t, X_t, Y_t are both parents of Z_t, since we assume both the sentence observation and the hidden state representation determine the sequence label. This DDN model maintains the same inference complexity as the HMM, since only the state sequence Y is latent during training. While by allowing bi-directional arcs over the label sequence Z, it has a natural capacity of modeling and incorporating task-specific label information for representation learning. By incorporating the label sequence Z into the model, we expect to identify more task discriminative latent sequence representations.

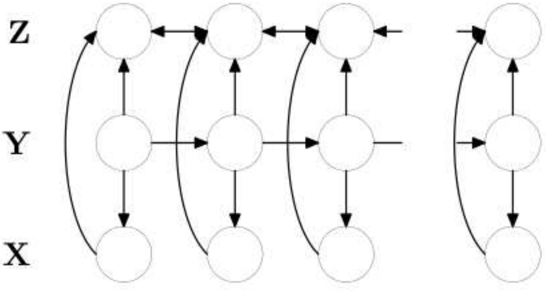

Figure 1: A Dynamic Dependency Network (DDN)

4.1 Training and Inference

Although we have an additional bi-directional Z layer in DDNs, the structures over the hidden layer Y, and between Y and the observed sequence X are similar to in HMMs. Thus inference over the hidden states and parameter learning in DDNs are as tractable as in HMMs. Assume that we are given a data set of N i.i.d. samples, $\{(X^i, Z^i)\}$ for $i = 1, 2, \ldots, N$, where X^i is the ith sentence and Z^i is the corresponding sequence of labels, e.g., POS tags, for X^i. Given the training data, its log-likelihood is

$$L(\theta) = \sum_{i=1}^{N} \log P(X^i, Z^i | \theta)$$

where θ denotes the set of model parameters.

Let $q(Y)$ be any non-zero distribution over hidden variables Y, we can get a lower bound for $L(\theta)$. For notational convenience, we will drop the superscript i in the following formulas.

$$\begin{aligned} \ell(\theta) &= \log \sum_{Y} q(Y) \frac{P(X,Y,Z|\theta)}{q(Y)} \qquad (1) \\ &\geq \sum_{Y} q(Y) \log \frac{P(X,Y,Z|\theta)}{q(Y)} \\ &= L(\theta) - D_{KL}(q(Y) \,\|\, P(Y|X,Z,\theta)) \qquad (2) \end{aligned}$$

where $D_{KL}(\cdot)$ denotes the Kullback-Leibler divergence measure. We denote the objective in (2) as $F(q,\theta)$. We then conduct training by maximizing $F(q,\theta)$ using iterative Expectation-Maximization (EM) algorithm (Baum et al., 1970; Dempster et al., 1977). For the $(k+1)$th iteration, in the E-step, we update q given fixed θ^k from previous iteration by

$$q^{k+1} = \arg\max_{q} F(q, \theta^k) \qquad (3)$$

which has the following solution when the KL divergence becomes zero

$$q^{k+1}(Y) = P(Y|X,Z,\theta^k). \qquad (4)$$

In the M-Step, we update θ given fixed q^{k+1}

$$\theta^{k+1} = \arg\max_{\theta} F(q^{k+1}, \theta) \qquad (5)$$

Similar to HMMs, the parameter estimation in (5) requires computation of $P(Y_{t-1}, Y_t | X, Z)$ and $P(Y_t|X,Z)$ for all t in the E-step. We extend the Baum-Welch algorithm used in HMMs to conduct the required computation with the current model parameters θ. Let $\alpha_t(y) = P(X_1, Z_1, \cdots, X_t, Z_t, Y_t = y|\theta)$ and $\beta_t(y) = P(X_{t+1}, Z_{t+1}, \cdots, X_T, Z_T | Y_t = y, \theta)$. The set of $\{\alpha_t(y)\}$ and $\{\beta_t(y)\}$ can be solved inductively using a forward procedure and a backward procedure respectively, which are analogous to the forward and backward procedures used for HMMs. Then the marginal probabilities can be computed as

$$P(Y_t = y|X,Z,\theta) = \frac{\alpha_t(y)\beta_t(y)}{\sum_{\widehat{y}} \alpha_T(\widehat{y})} \qquad (6)$$

$$P(Y_t = y, Y_{t+1} = y\prime | X, Z, \theta) = \tag{7}$$
$$\frac{\alpha_t(y)\beta_{t+1}(y\prime)}{\sum_{\hat{y}} \alpha_T(\hat{y})} P(Y_{t+1} = y\prime | Y_t = y) P(Z_{t+1} | Z_t, Z_{t+2}, X_{t+1}, Y_{t+1} = y\prime)$$

The major difference from HMMs is that the computation of (7) requires the additional local probabilities, $P(Z_t|Z_{t-1}, Z_{t+1}, X_t, Y_t)$, in the bi-directional Z sequence. The typical conditional probability table (CPT) parameters for $P(Z_t|Z_{t-1}, Z_{t+1}, X_t, Y_t)$ requires a storage space in the size of $(L-1) \times L^2 \times V \times S$, where L is the number of discrete label values for Z, V is the number of discrete word features for X, and S is the number of discrete states for Y. To reduce the computational cost and memory size for storing such a large CPT and increase the scalability of the proposed model, we exploit a multi-class logistic regression model to model this conditional probability distribution and store the model parameters of the logistic regression model instead.

The logistic regression classifier is trained in the M-step with data collected at each location t, over four types of features $Z_{t-1}, Z_{t+1}, X_t, Y_t$. Given the model parameters θ, the hidden state values of sequence Y are computed using the Viterbi inference algorithm used in HMMs. Thus the trained logistic regression model only requires a model parameter matrix W in the size of $L \times (2L + V + S + 1)$ to calculate the probability $P(Z_t|Z_{t-1}, Z_{t+1}, X_t, Y_t, W)$ for any inputs. Apparently, the space required to store the W matrix is much smaller than the space required for the original conditional probability table. To avoid overfitting, we trained a $L2$-norm regularized logistic regression model using a second-order Newton method.

With the computed marginal probabilities and induced hidden states, the model parameters θ of the DDN can be re-estimated in a similar way as in HMMs in addition to the retraining of the logistic regression classifier.

4.2 Semi-supervised Representation Learning

We have introduced above how to train DDNs with labeled sentences and conduct inference to induce the hidden states. For cross-domain semi-supervised representation learning, we have a small amount of labeled sentences $\{(X^l, Z^l)\}$ in the source domain and a large amount of unlabeled sentences $\{X^u\}$ in both the source and target domains. This requires the DDN model to handle unlabeled sentences as well. Note in the DDN model we introduced, dropping the label layer Z does not affect either the structure nor the parameter of the other two layers, but simplify a DDN model into a HMM. Thus we can use DDNs as HMMs on unlabeled sentences by sharing common model parameters across labeled and unlabeled sentences. With this semi-supervised representation learning, we expect to inference latent features that are not only generalizable in different domains, but also more informative or discriminative about the target task labels.

Our overall system follows a similar procedure of (Huang and Yates, 2009). First we train a DDN model over both the labeled sentences in the source domain and the unlabeled sentences in both domains, as we described above. Then we use the trained DDN model to produce latent features (i.e, hidden state values Y) for the training and test sentences using Viterbi inference algorithm. Finally we train a classification model, e.g., CRFs, over the training sentences for the target task, e.g. POS tagging, using the latent features as augmented inputs, and then perform classification on the test sentences. We expect semi-supervised representation learning to help improve out-of-domain prediction performance with more discriminative latent features.

5 Domain Adaptation for Part-of-Speech Tagging

In this section, we report our empirical study on how semi-supervised representation learning can improve out-of-domain part-of-speech tagging accuracy.

5.1 Datasets

We used the same datasets as (Blitzer et al., 2006; Huang and Yates, 2009, 2010). The source domain contains articles from Wall Street Journal (WSJ), with 39,832 manually tagged sentences from sections 02-21 and 100,000 unlabeled sentences from a 1988 subset. The target domain contains bio-medical articles from MEDLINE, with 561 labeled sentences[1] and 100,000 unlabeled sentences. The task is to assign words with one of the POS tags from the Penn Treebank POS tags (Marcus et al., 1993) and two more tags from MEDLINE dataset. Among the tags, two tags cannot be seen in the newswire articles, *HYPH* (hyphens) and *AFX* (common post-modifiers for biomedical entities such as genes). These two tags were introduced because of the importance of hyphenated entities in biomedical text, which are about 1.8% of the words in the 561 labeled sentences.

5.2 Representation Learning

We explored both unsupervised representation learning using HMMs and semi-supervised representation learning using the proposed DDNs. We built a vocabulary with all sentences from the source and target domains. In order to reduce the vocabulary size, we further applied the preprocessing steps used in (Huang and Yates, 2009, 2010): we mapped lower frequency (0-2) words to a single unique identifier and sole-digit words into a single unique identifier in our vocabulary. With these preprocessed sentences, we applied representation learning models (DDNs and HMMs) to derive hidden states as additional features for supervised POS taggers.

We used HMMs to perform unsupervised representation learning on 139,832 newswire sentences and 100,000 unlabeled biomedical sentences following the work (Huang and Yates, 2009). Then we decoded the hidden states for 39,832 newswire sentences (the labeled sentences in the source domain) as well as 561 biomedical sentences (the test sentences in the target domain) as additional features for supervised POS tagging. In the unsupervised representation training, one hyperparameter, the number of hidden states, has to be set. A large number of hidden states would make the model more capable to derive latent features, however, it also needs more memory storage and high computation cost. We used 80 states in our experiments, following (Huang and Yates, 2009), to produce fair comparisons.

We used the proposed DDN model for semi-supervised representation learning on 39,832 labeled and 100,000 unlabeled newswire sentences as well as 100,000 unlabeled biomedical sentences. The labels we used in semi-supervised representation learning are the same labels we will use later to train POS taggers. Thus comparing to unsupervised representation learning, the semi-supervised representation learning does not require additional annotation effort, but makes use of the existing labels in the source domain. For our semi-supervised representation learning, we need to choose two hyperparameters, the number of hidden states and the L2 regularization parameter. We set the former as 80, same as in unsupervised representation learning. Our model is not sensitive to the L2 regularization parameter and we set it as 0.5.

[1] Sentences are manually annotated as part of the Penn BioIE project.

5.3 Part-of-Speech Tagging Accuracy

For supervised POS tagging, the training data contains 39,832 labeled newswire sentences and the test data contains 561 biomedical sentences. The 561 biomedical sentences contain 14,554 tokens, of which 23% are OOV (Out-Of-Vocabulary) tokens. We tested our semi-supervised representation learning using supervised Conditional Random Field (CRF) POS taggers and used a fast-training CRF package developed by Okazaki (2007). The feature set used for the CRF POS tagger is presented in Table 1. Specifically, we extracted unigram features. We also added orthographical features such as suffix (-ing, -ogy, -ed, -ly, -s, -ion, -tion, -ity), as well as capitalization. Orthographical features contribute to improving tagging accuracy for out-of-vocabulary words as is demonstrated by (Lafferty, 2001). In addition, we added the latent states as state features for each word from the learned representations. For example, a sentence like "He is the CEO ." contains 5 words: 4 regular words and a "period". A state feature is learned for each of them.

Table 1: CRF feature set used in our supervised CRF POS taggers. Z_i variables stand for labels to be predicted, W_is represent word tokens. Y_is stand for hidden state values decoded from HMM or DDN models, i.e., the new representation features.

Feature Type	Feature Description
Transition	$Z_i = t$ $Z_i = t$ and $Z_{i-1} = t'$
Word	$W_i = w$ and $Z_i = t$
Orthography	For every s ∈ {-ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity}, suffix(W_i)=s and $Z_i = t$ W_i is capitalized and $Z_i = t$ W_i has a digit and $Z_i = t$
HMM features	$Z_i = t$ and $Y_i = y$
DDN features	$Z_i = t$ and $Y_i = y$

Our experimental results in term of per-token accuracy with different representation learning methods are presented in Table 2. For all test results reported in this paper, the "*All Words*" results are average accuracies over all words in the test data, the "*OOV Words*" results are average accuracies over only OOV words in the test data that appeared less than 3 times in the training data. We reported the empirical results for the following approaches:

- **Baseline:** the baseline CRF POS-tagger trained without representation learning.
- **ASO:** the Alternating Structural Optimization technique in (Ando and Zhang, 2005).
- **SELF-CRF:** the comparison method using a self-training paradigm. We first train a CRF without representation learning on the training data and apply it on the test data, then retrain it on the training data plus the test data with predicted labels.
- **PLAIN-SEM:** the method based on the representation learning technique using contrastive estimation (Smith and Eisner, 2005). We used the modified version in (Huang and Yates, 2010).

Table 2: Per-token accuracy for out-of-domain words on MEDLINE domain trained with Wall Street Journal articles.

Approaches	All Words	OOV Words
Baseline	88.3%	67.3%
ASO	88.4%	70.9%
SELF-CRF	88.5%	70.4%
PLAIN-SEM	88.5%	69.8%
SCL	88.9%	72.0%
SEM-CRF	90.0%	71.9%
HMM	90.5%	75.2%
DDN	**91.3%**	**76.1%**

- **SCL:** the method based on the representation learning with the Structural Correspondence Learning (SCL) technique, developed by (Blitzer et al., 2006).
- **SEM-CRF:** the method based on the representation learning in (Huang and Yates, 2010).
- **HMM:** the method based on the unsupervised representation learning using HMMs in (Huang and Yates, 2009).
- **DDN:** the method based on the proposed semi-supervised representation learning.

We also investigated how our representation learning benefits supervised POS taggers by varying the number of labeled training sentences from the source domain. For comparison, we considered *Baseline*, *SCL* and *HMM*, since *SCL* and *HMM* work very well among all the other comparison methods. The per-token accuracies on test data are reported in Figure 2.

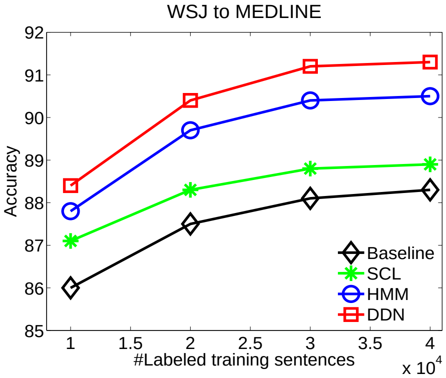

Figure 2: Per-token accuracies for out-of-domain POS tagging. WSJ is used as the source domain and MEDLINE is used as the target domain.

From Table 2 and Figure 2, we can see that with semi-supervised representation learning, DDN consistently outperforms other comparison methods for out-of-domain POS tagging. From Figure 2, we can see that by increasing the number of labeled training data, *DDN* can gain

Table 3: Statistical Significance (McNemar's) tests for out-of-domain experiments with CRF POS taggers. Results are significant with $p < 0.05$.

Null Hypothesis	p-value
HMM vs. Baseline	2.3×10^{-9}
DDN vs. Baseline	3.4×10^{-10}
DDN vs. SCL	6.7×10^{-7}
DDN vs. HMM	2.9×10^{-4}

more improvements in accuracy compared with *Baseline*, *SCL* and *HMM*. Specifically, *DDN* increases accuracy by 2.4% compared with *Baseline*, by 1.3% compared with *SCL*, and by 0.6% compared with *HMM* when the labeled training data is 1,000. When the labeled training data reaches 39,832, *DDN* increases accuracy by 3.0% compared with *Baseline*, by 2.4% compared with *SCL*, and by 0.8% compared with *HMM*. Those results suggest that *DDN* can produce more effective task-specific features by incorporating existing labels from the source domain, and further assist out-of-domain POS tagging.

We also present results for corresponding significance tests over comparisons between *Baseline*, *SCL*, *HMM* and *DDN* in Table 3. We followed the experiments in (Blitzer et al., 2006), and used a McNemar paired test for labeling disagreements (Gillick and Cox, 1989) with $p < 0.05$ being significant on all test words. We report the p values in Table 3. We can see that *DDN* significantly improves out-of-domain tagging accuracy over *Baseline*, *SCL* and *HMM*.

6 Domain Adaptation for Syntactic Chunking

In this section, we empirically study how our proposed semi-supervised representation learning can improve out-of-domain performance on syntactic chunking.

6.1 Datasets

We used the datasets from the CoNLL 2005 shared task (Carreras and Màrquez, 2005) for our second set of experiments on syntactic chunking. We used the standard training set, consisting of sections 02-21 of the Wall Street Journal (WSJ) portion of the Penn Treebank, and conducted tests on the Brown corpus (Kucera and Francis, 1967). The test data contains 3 sections (ck01-ck03) of propbanked Brown corpus data, which consists of 426 sentences containing 7,159 tokens. Besides these labeled data, we also incorporated unlabeled data from both domains. We added 100,000 unlabeled news sentences for the source domain and 57,000 unlabeled sentences for the target domain. In this setting, while the source domain contains newswire text, the test sentences are drawn from the domain of "general fiction" and contain entirely different styles of English.

The original training data and test data from the CoNLL 2005 shared task contain POS tags as well as partial syntax, namely chunks and clauses. In order to perform syntactic chunking task, we mapped the partial syntax labels to chunking labels in IOB2 format. IOB2 format is a standard format for various sequence tasks like syntactic chunking and it is widely used in previous works including the CoNLL 2000 shared task[2]. In IOB2 format, the chunk tags

[2] http://www.clips.ua.ac.be/conll2000/chunking/.

Table 4: Average chunking performance on Brown corpus, with Wall Street Journal articles as training data.

Methods	F1
Baseline	89.93%
SELF-CRF	90.21%
SCL	90.62%
HMM	91.79%
DDN	**93.05**%
UPC Chunker	91.73%

consist of two parts. The first part represents the position of the token in this chunk and the second part stands for the name of the chunk type. For example, the chunking type of *VP* is used for verb phrase words and the chunking type of *NP* is used for noun phrase words. For words forming a chunk of type k, the first word receives the B-k tag (Begin), and the remaining words receive the tag I-k (Inside). Words outside a chunk receive the tag O. Below we give an example of a sentence labeled with chunking tags in IOB2 format from the source domain:

> The/B-NP $/I-NP 1.4/I-NP billion/I-NP robot/I-NP spacecraft/I-NP faces/B-VP a/B-NP six-year/I-NP journey/I-NP to/B-VP explore/I-VP Jupiter/B-NP and/O its/B-NP 16/I-NP known/I-NP moons/I-NP ./O

6.2 Representation Learning

We built a vocabulary with all sentences from the source and target domains. In order to reduce the vocabulary size, we used the same preprocessing steps as in POS tagging experiments, mapping lower frequency (0-2) words to a single unique identifier in our vocabulary and sole-digit words into a single unique identifier. On the preprocessed sentences, we then applied representation learning models (DDNs or HMMs) to derive hidden states of the sentence words, which can be used as additional features for supervised syntactic chunking systems.

We used HMMs to perform unsupervised representation learning on 139,832 newswire source sentences and 57,000 unlabeled "general fiction" sentences from Brown corpus. Then we decoded the hidden states for 39,832 labeled newswire sentences and 426 "general fiction" test sentences as additional features for supervised syntactic chunking, using the trained HMM. In the unsupervised representation training, one hyperparameter, the number of hidden states, has to be set. We used 80 states in our experiments in consideration of the model capability, memory storage and computation cost.

We used the proposed DDN model for semi-supervised representation learning on the same data as for HMMs, i.e., 139,832 newswire sentences from the source domain and 57,000 unlabeled "general fiction" sentences from the target domain. But different from unsupervised representation learning, our proposed semi-supervised representation learning makes use of the existing labels of the 39,832 sentences in the source domain. In our semi-supervised representation learning, we need to choose two hyperparameters, the number of hidden states and the L2 regularization parameter. We set the former as 80 and the latter as 0.5, which are same as in our previous experiments for POS-tagging.

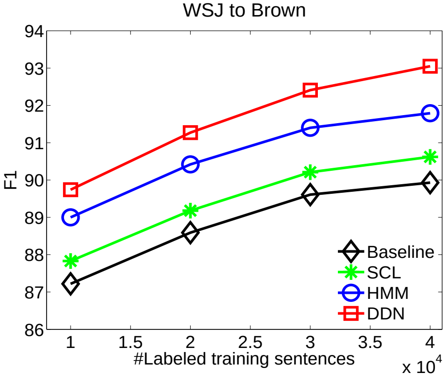

Figure 3: Results in term of F1 measure for out-of-domain syntactic chunking. WSJ is used as the source domain and Brown corpus is used as the target domain.

6.3 Syntactic Chunking Results

For supervised syntactic chunking, the training data contains 39,832 labeled newswire sentences and the test data contains 426 "general fiction" sentences. The 426 "general fiction" sentences contain 7,159 tokens. We tested our semi-supervised representation learning with supervised Conditional Random Field (CRF) syntactic chunking. We used the same fast-training CRF package developed by Okazaki (2007). For syntactic chunking, in addition to the CRF feature set in Table1 which we used in POS tagging experiments, we also extracted POS tag features. All features are represented with boolean values.

Our experimental results with different representation learning methods are presented in Table 4. The results are in term of F1 measure, since F1 measure is widely used in syntactic chunking tasks (Huang and Yates, 2009; Carreras and Màrquez, 2005). We reported the empirical results of the following approaches for comparison:

- **UPC Chunker:** a chunking system based on Voted Perceptrons (Carreras and Màrquez, 2003). Carreras and Màrquez (2005) trained such a chunker on WSJ sections 02-21 and tested it on three sections of the Brown corpus (ck01-03). The reported results serve as the current *state-of-the-art* performance on this experimental setting.
- **Baseline:** the baseline CRF chunker without representation learning.
- **SELF-CRF:** the CRF chunker with a self-training paradigm. We first train a CRF without representation learning on the training data and apply it to the test data, then retrain it on the training data plus the test data with predicted labels.
- **SCL:** the method based on the representation learning produced using the Structural Correspondence Learning (SCL) technique (Blitzer et al., 2006).
- **HMM:** the method based on unsupervised representation learning using Hidden Markov Models (Huang and Yates, 2009).
- **DDN:** the method based on the proposed semi-supervised representation learning.

We also investigated the performance of the proposed DDN-based chunker by varying the number of labeled training sentences from the source domain. Its comparison results with *Baseline*, *SCL* and *HMM* are presented in Figure 3.

From Table 4 and Figure 3, we can see that with semi-supervised representation learning, the DDN based chunker consistently outperforms other methods for out-of-domain syntactic chunking. According to Figure 3, by increasing the number of labeled training data, *DDN* can gain more improvements in term of F1 measure comparing to *Baseline*, *SCL* and *HMM*. Specifically, *DDN* increases the F1 by 2.52% comparing with *Baseline*, by 1.91% comparing with *SCL*, and by 0.74% comparing with *HMM* when the number of labeled training sentences is 1,000. When the number of labeled training sentences reaches 39,832, *DDN* outperforms *Baseline* by 3.08%, outperforms *SCL* by 2.43%, and outperforms *HMM* by 1.26%, in term of F1-measure. These results again suggest that the semi-supervised representation learning method, *DDN*, can produce more effective task-specific features by incorporating existing labels from the source domain.

We also produced the results of corresponding significance tests, reported in Table 5. We used a McNemar paired test for labeling disagreements (Gillick and Cox, 1989) with $p < 0.05$ being significant on all test words. We reported the p values in Table 5, from which we can see that *DDN* significantly improves out-of-domain chunking performance over *Baseline*, *SCL* and *HMM*.

Table 5: Statistical Significance (McNemar's) tests for out-of-domain experiments with CRF syntactic chunkers. Results are statistical significant with $p < 0.05$.

Null Hypothesis	p-value
HMM vs. Baseline	5.6×10^{-8}
DDN vs. Baseline	2.9×10^{-10}
DDN vs. SCL	7.1×10^{-8}
DDN vs. HMM	4.7×10^{-4}

Conclusion

In this paper, we proposed a Dynamic Dependency Network model for semi-supervised representation learning. In addition to the large amount of unlabeled data from two domains, it incorporates the task-specific labels from the source training data into representation learning. We then used the induced generalizable state features to augment source training sentences and target test sentences for two cross domain NLP tasks: part-of-speech tagging and syntactic chunking. Our empirical studies show that the proposed semi-supervised representation learning outperforms unsupervised representation learning based on HMMs on out-of-domain test data for both POS tagging system and syntactic chunking system. With the proposed semi-supervised representation learning, the POS taggers and the syntactic chunkers resulted also outperform a set of other POS tagging methods and syntactic chunking methods for out-of-domain predictions. All results suggest the proposed semi-supervised representation learning can better bridge the domain gap between training sentences and test sentences by exploiting task-specific label information in the representation learning process.

Acknowledgments

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

References

Ando, R. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research (JMLR)*, 6:1817–1853.

Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. (2010). A theory of learning from different domains. *Machine Learning*, 79:151–175.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Blitzer, J., Foster, D., and Kakade, S. (2011). Domain adaptation with coupled subspaces. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Carreras, X. and Màrquez, L. (2003). Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*.

Carreras, X. and Màrquez, L. (2005). Introduction to the conll-2005 shared task: semantic role labeling. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

Daumé III, H., Kumar, A., and Saha, A. (2010). Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Dredze, M., Kulesza, A., and Crammer, K. (2010). Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149.

Gillick, L. and Cox, S. (1989). Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., and Kadie, C. (2000). Dependency networks for inference, collaborative filtering and data visualization. *Journal of Machine Learning Research (JMLR)*, 1:49–75.

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

Huang, F. and Yates, A. (2010). Exploring representation-learning approaches to domain adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Kucera, H. and Francis, W. (1967). *Computational analysis of present-day American English*. Brown University Press.

Lafferty, J. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of english: The penn treebank. In *Proceedings of the Annual Meeting of Association of Computational Linguistics (ACL)*.

McClosky, D., Charniak, E., and Johnson, M. (2010). Automatic domain adaptation for parsing. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL)*.

Mnih, A. and Hinton, G. (2009). A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems (NIPS)*.

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis.

Okazaki, N. (2007). CRFsuite: a fast implementation of conditional random fields (CRFs). http://www.chokkan.org/software/crfsuite/.

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

Sekine, S. (1997). The domain dependence of parsing. In *Proceedings of the Conference on Applied natural language processing*.

Smith, N. and Eisner, J. (2005). Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Turian, J., Ratinov, L., and Bengio, Y. (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)*.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc.