Domain Adaptation for Sequence Labeling Tasks with a Probabilistic Language Adaptation Model

Min Xiao MINXIAO@TEMPLE.EDU
Yuhong Guo YUHONG@TEMPLE.EDU
Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA

Abstract

In this paper, we propose to address the problem of domain adaptation for sequence labeling tasks via distributed representation learning by using a log-bilinear language adaptation model. The proposed neural probabilistic language model simultaneously models two different but related data distributions in the source and target domains based on induced distributed representations, which encode both generalizable and domain-specific latent features. We then use the learned dense real-valued representation as augmenting features for natural language processing systems. We empirically evaluate the proposed learning technique on WSJ and MEDLINE domains with POS tagging systems, and on WSJ and Brown corpora with syntactic chunking and named entity recognition systems. Our primary results show that the proposed domain adaptation method outperforms a number of comparison methods for cross domain sequence labeling tasks.

1. Introduction

Domain adaptation aims to learn a prediction model for a label-scarce *target* domain by exploiting information in a label-rich *source* domain (Blitzer et al., 2006; Daumé III, 2007; Ben-David et al., 2007; Daumé III et al., 2010). Domain adaptation is prevailingly needed for various sequence labeling tasks in natural language processing (NLP) area, such as syntactic chunking (Daumé III, 2007; Huang & Yates, 2009), part-of-speech (POS) tagging (Blitzer et al., 2011; 2006), parsing (McClosky et al., 2010), semantic role labeling (Carreras & Màrquez, 2005), and named entity recognition (NER) (Daumé III, 2007; Turian et al., 2010; Daumé III & Marcu, 2006).

In a practical domain adaptation learning scenario in NLP, the source domain and the target domain usually have *very different vocabularies*, which renders the lexical feature-based NLP systems to perform poorly on the target domain (Ben-David et al., 2007; 2010). For example, a statistical machine learning model for POS tagging systems based on lexical features trained on newswire data with frequent terms like "CEO", "corporation" cannot correctly infer POS tags for biomedical text with frequent terms like "metastases", "sequencing" and "genomic". Moreover, the learning machine based on lexical features may produce *inconsistent predictions* across domains. For example, the word "signaling" in "signaling that ..." from the Wall Street Journal (WSJ) domain is a verb (VBG), but it is a noun (NN) in "signaling pathway" from the MEDLINE domain (Huang & Yates, 2010). Recently, much work has been proposed to cope with those problems in order to improve the prediction performance for out-of-domain NLP systems, including feature augmentation based supervised adaptation method (Daumé III, 2007), semi-supervised adaptation method (Daumé III et al., 2010), and representation learning methods (Blitzer et al., 2006).

In this paper, we propose to adapt sequence labeling systems in NLP from a source domain to a different but related target domain by inducing distributed representations using a log-bilinear language adaptation (LBLA) model. It combines the advantages of representation learning methods from (Blitzer et al., 2006), which employ generalizable features across domains to reduce domain divergence, and feature augmentation based (semi-)supervised adaptation learning methods from (Daumé III, 2007; Daumé III et al., 2010), which exploit both common and domain-specific features from both domains. Specifically, the LBLA model simultaneously models the source distribution by learn-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

ing generalizable and source-specific word representations and models the target distribution by learning domain-sharing and target-specific word representations. We then use the learned representation embedding functions to map the original data into the induced representation space as augmenting features, which are incorporated into supervised sequence labeling systems to enable cross domain adaptability. The proposed learning technique is empirically evaluated for cross domain POS tagging systems on sentences from WSJ and MEDLINE domains, cross domain syntactic chunking and named entity recognition systems on sentences from WSJ and Brown corpora, and is shown to outperform a number of related methods.

2. Related Work

Domain adaptation has been intensively studied for a variety of sequence labeling tasks in the natural language processing area. Daumé III & Marcu (2006) proposed to distinguish between general features and domain-specific features by training three separate maximum entropy classifiers. They empirically showed the effectiveness of the proposed method on mention type classification, mention tagging and recapitalization systems. Jiang & Zhai (2007) investigated instance weighting method for semi-supervised domain adaptation by assigning more weights to labeled source and target data, removing misleading training instances in the source domain, and augmenting target training instances with predicted labels. They empirical evaluated their method for cross domain part-of-speech tagging and named entity recognition to justify its efficacy. Daumé III (2007) proposed an easy adaptation learning method (EA) by using feature replication, which is later extended into a semi-supervised version (EA++) by incorporating unlabeled data via co-regularization (Daumé III et al., 2010). These methods demonstrated good empirical performance on a variety of NLP tasks.

Recently, *representation learning* methods have been proposed to induce generalizable features by exploiting large amount of unlabeled data from both domains, which are then used to augment original instances to improve cross domain prediction performance (Blitzer et al., 2006; Ando & Zhang, 2005; Huang & Yates, 2009; 2010). Blitzer et al. (2006) proposed to seek for common latent features by performing structural correspondence learning (SCL), which models the correlation between pivots (frequent lexical features) and non-pivot features. Huang & Yates (2009) proposed to induce hidden states as latent features by training Hidden Markov Models (HMMs) on unlabeled sentences from two domains. They empirically demonstrated the efficacy of their approach on out-of-domain part-of-speech tagging and syntactic chunking tasks. Their learning technique is also further exploited in (Huang & Yates, 2010), which aims to learn a multi-dimensional feature representation by simultaneously training multiple HMMs with different initializations. Turian et al. (2010) empirically demonstrated that employing Collobert and Weston embeddings (Collobert & Weston, 2008), Brown clusters, or HLBL embeddings (Mnih & Hinton, 2009) as extra word features can improve the performance of out-of-domain named entity recognition systems and in-domain syntactic chunking systems.

Distributed representations are widely exploited in natural language processing area. Bengio et al. (2000; 2003) introduced neural network language models and demonstrated how to combine neural network probability predictions with distributed representations for words in order to outperform standard n-gram models. Blitzer et al. (2004) demonstrated that those learned distributed representations of symbols make sense linguistically. The effectiveness of distributed representations has also been demonstrated on other NLP tasks, such as sentiment analysis (Maas & Ng, 2010), syntactic chunking, named entity recognition(Turian et al., 2010), semantic role labelling (Collobert & Weston, 2008), and parsing (Socher et al., 2011).

3. Proposed Approach

Previous empirical results showed that latent generalizable features can increase the accuracy for out-of-domain prediction performance (Blitzer et al., 2006). It has also been justified by a recent theoretic study that a proper feature representation is crucial to domain adaptation due to its contribution on bridging domain divergence (Ben-David et al., 2007; 2010). In this work, we propose to learn generalizable distributed representations of words from sentence structures to address the problem of domain adaptation for sequence labeling tasks in NLP.

Distributed representations, which are dense, low-dimensional, and continuous-valued, are called word embeddings (Turian et al., 2010). Each dimension of the word embedding stands for a latent feature of the word, hopefully capturing useful semantic and syntactic regularities. The basic idea to learn a distributed representation is to link each word with a real-valued feature vector, typically by using neural language models. A sentence can thus be transformed into a sequence of these learned feature vectors. The neural language model learns to map the sequence of feature

vectors to a prediction of interest, such as the conditional probability distribution over the current word given its previous context, and pushes the learned word features to form grammatical and semantic similarities (Bengio et al., 2000; 2003). The advantage of this distributed representation method is that it allows the model to generalize well to sequences that do not appear in the training set, but are similar to training sequences with respect to their distributed representations (Bengio et al., 2000). The simplest neural language model is the log-bilinear language (LBL) model developed in (Mnih & Hinton, 2007), which performs linear predictions in the semantic word feature space. Despite its simplicity, the LBL model has been shown to outperform n-grams on a large dataset (Mnih & Hinton, 2007; Mnih & Teh, 2012). Based on this simple language model, we present a log-bilinear language adaptation (LBLA) model below to learn adaptive distributed word representations for domain adaptation over sequence labeling tasks.

3.1. Log-Bilinear Language Adaptation Model

We consider the domain adaptation problem from a source domain $\mathcal{S}$ to a target domain $\mathcal{T}$. In the source domain, we have l_s labeled sentences $\{(X_i^s, Y_i^s)\}_{i=1}^{l_s}$ and u_s unlabeled sentences $\{(X_i^s)\}_{i=l_s+1}^{n_s}$ for $n_s = l_s + u_s$, where X_i^s is the ith input sentence, i.e., a sequence of words, $w_1, w_2, \ldots, w_{T_i}$, and Y_i^s is its corresponding label sequence, e.g. the sequence of POS tags. Similarly, in the target domain, we have l_t labeled sentences $\{(X_i^t, Y_i^t)\}_{i=1}^{l_t}$ and u_t unlabeled sentences $\{(X_i^t)\}_{i=l_t+1}^{n_t}$ for $n_t = l_t + u_t$. Typically, l_t is much smaller than l_s. Though the two domains may have very different vocabularies, for simplicity we use a common word vocabulary V for both domains.

We adapt the log-bilinear language model to learn distributed representations across domains by simultaneously modeling two different but related data distributions in the source and target domains. The distributed representations are encoded as real-valued vectors for words in the vocabulary. We refer to the matrix with all word representation vectors as R and denote the representation vector for word w as $R(w)$. Motivated by (Daumé III, 2007), we split the representation vector into three parts to capture both domain-sharing and domain-specific properties of each word. Thus the representation vector for a word w can be expressed as

$$R(w) = [R^s(w); R^c(w); R^t(w)] \tag{1}$$

where $R^s(w)$ represents source-specific latent features, $R^c(w)$ represents common latent features, and $R^t(w)$ represents target-specific latent features. Naturally we assume the source domain contains no target-specific features and the target domain contains no source-specific features. In practice, we define two mapping functions, Φ^s and Φ^t, to map the observed source and target words to cross domain word embeddings

$$\Phi^s(w) = [R^s(w); R^c(w); \mathbf{0}^t] \tag{2}$$

$$\Phi^t(w) = [\mathbf{0}^s; R^c(w); R^t(w)] \tag{3}$$

where $\mathbf{0}^t$ is a zero vector in the same size of $R^t(w)$ and $\mathbf{0}^s$ is a zero vector in the same size of $R^s(w)$. Note that Φ^s and Φ^t are different from those mapping functions exploited in previous work on domain adaptation (Daumé III, 2007), since we propose to learn the latent feature vectors using a log-bilinear language model while they perform simple feature replication. The common and domain-specific features learned for a word w can be different in our proposed model, while they use two identical copies for both parts.

The three-part distributed representation learning framework can explicitly model the relationship between two data sources through the common representation part, while still maintaining the unique semantic and syntactic information of each data source through the domain-specific parts. For example, a POS tagging task uses WSJ as the source domain and MEDLINE as the target domain. The word "signaling" in a sentence "signaling that ..." from the source domain is a verb (VBG), but it is a noun (NN) in a sentence "signaling pathway ..." from the target domain. This syntactic difference of the same word in two domains can be encoded in the domain-specific latent features in the distributed representation.

Recall that we have two sets of training sentences sampled from two domains, $\mathcal{S}$ and $\mathcal{T}$. The LBLA model thus includes a set of conditional distributions, $P_{\mathcal{D}}(w|h)$ of each word w given its previous $(n_c - 1)$ words (denoted as the context h), for each domain $\mathcal{D} \in \{\mathcal{S}, \mathcal{T}\}$,

$$P_{\mathcal{D}}(w|h;\theta) = \frac{\exp\left(-E_{\mathcal{D}}(w,h;\theta)\right)}{Z_{\mathcal{D}}(h;\theta)} \tag{4}$$

$$Z_{\mathcal{D}}(h;\theta) = \sum_{w'} \exp\left(-E_{\mathcal{D}}(w',h;\theta)\right).$$

Here $E_{\mathcal{D}}(w,h;\theta)$ is a log-bilinear energy function,

$$E_{\mathcal{D}}(w,h;\theta) = -\widehat{\Phi}^d(w)^T \Phi^d(w) - b_w \tag{5}$$

for $d \in \{s,t\}$ correspondingly, and it quantifies the compatibility of word w with context h in domain $\mathcal{D}$. b_w is the bias parameter used to capture the popularity of word w across domains. We refer to the bias vector for all words as $\mathbf{b}$. $\widehat{\Phi}^d(w)$ is the predicted representation vector for the target word w given its context h in

the domain indexed by d, which can be computed by linearly combining the feature vectors for the context words, such that

$$\widehat{\Phi}^s(w) = \sum_{i=1}^{n_c-1} \left[C_i^s R^s(w_i); C_i^c R^c(w_i); \mathbf{0}^t\right] \quad (6)$$

$$\widehat{\Phi}^t(w) = \sum_{i=1}^{n_c-1} \left[\mathbf{0}^s; C_i^c R^c(w_i); C_i^t R^t(w_i)\right] \quad (7)$$

where C_i^s, C_i^c, C_i^t are the position-dependent context weight matrices for source-specific, common and target-specific features respectively. Thus the negated log-bilinear energy function $-E_{\mathcal{D}}(w, h; \theta)$ measures the similarity between the current word feature vector $\Phi^d(w)$ and the predicted feature vector $\widehat{\Phi}^d(w)$.

Overall, the proposed LBLA model simultaneously models two sets of conditional distributions, $P_{\mathcal{S}}(.;\theta)$ and $P_{\mathcal{T}}(.;\theta)$, respectively on the two domains based on distributed feature representations. These two sets of distributions reflect both domain-sharing properties of the data, parameterized with $\{R^c, \{C_i^c\}, b_w\}$, and domain-specific properties of the data, parameterized with $\{R^s, \{C_i^s\}\}$ and $\{R^t, \{C_i^t\}\}$.

3.2. Training with Noise-Contrastive Estimation

We propose to train the LBLA model using noise-contrastive estimation (NCE) (Gutmann & Hyvärinen, 2010; 2012), which has been recently introduced for training unnormalized probabilistic models, and is shown to be less expensive than maximum likelihood learning and more stable than importance sampling (Bengio & Senécal, 2003) for training neural probabilistic language models (Mnih & Teh, 2012). Assume that we have a source data distribution $P_{\mathcal{S}}(w|h)$ and a target data distribution $P_{\mathcal{T}}(w|h)$, which are the distributions of words occurring after a particular context h, on the source domain and the target domain respectively. We propose to distinguish the observed source data and the observed target data from noise samples, which are artificially generated by a unigram distribution. We denote the context-independent noise distribution as $P_n(w)$. We would like to fit the context-dependent parameter models $P_{\mathcal{S}}(w|h;\theta)$ and $P_{\mathcal{T}}(w|h;\theta)$ to $P_{\mathcal{S}}(w|h)$ and $P_{\mathcal{T}}(w|h)$ respectively.

We assume that observed samples appear k times less frequently than noise samples, and data samples come from the mixture distribution

$$\frac{k}{k+1} P_n(w) + \frac{1}{k+1} P_{\mathcal{D}}(w|h) \quad (8)$$

on a domain $\mathcal{D}$, for $\mathcal{D} \in \{\mathcal{S}, \mathcal{T}\}$. Since we are fitting $P_{\mathcal{D}}(w|h,\theta)$ to $P_{\mathcal{D}}(w|h)$, we will replace $P_{\mathcal{D}}(w|h)$ with $P_{\mathcal{D}}(w|h;\theta)$. Then given a context h, the posterior probabilities that a sample word w comes from the observed source data distribution and the observed target data distribution are

$$P_{\mathcal{S}}(\mathrm{D}=1|w,h;\theta) = \frac{P_{\mathcal{S}}(w|h;\theta)}{kP_n(w) + P_{\mathcal{S}}(w|h;\theta)} \quad (9)$$

$$P_{\mathcal{T}}(\mathrm{D}=1|w,h;\theta) = \frac{P_{\mathcal{T}}(w|h;\theta)}{kP_n(w) + P_{\mathcal{T}}(w|h;\theta)} \quad (10)$$

However, evaluating Eq. (9) and Eq. (10) is too expensive due to the normalization computation for $P_{\mathcal{D}}(w|h;\theta)$ (Eq. 4). To tackle this issue, instead of conducting explicit normalization, NCE treats the normalization constants as parameters and parameterizes the models with respect to learned normalization parameters $z^s(h), z^t(h)$ and unnormalized distributions $P_{\mathcal{S}}(.|h;\theta^0)$, $P_{\mathcal{T}}(.|h;\theta^0)$, such that

$$\log P_{\mathcal{S}}(w|h;\theta) = \log P_{\mathcal{S}}(w|h;\theta^0) + z^s(h) \quad (11)$$

$$\log P_{\mathcal{T}}(w|h;\theta) = \log P_{\mathcal{T}}(w|h;\theta^0) + z^t(h) \quad (12)$$

where $\theta = \{\theta^0, z^s(h), z^t(h)\}$ and θ^0 are the parameters of the unnormalized distributions.

To fit the context-dependent model to the data, given a context h, we simply maximize an objective $J_{\mathcal{D}}(h;\theta)$ for each domain $\mathcal{D} \in \{\mathcal{S}, \mathcal{T}\}$. It is the expectation of $\log P_{\mathcal{D}}(\mathrm{D}|w,h;\theta)$ under the mixture distribution of the noise and observed data samples,

$$\begin{aligned} J_{\mathcal{D}}(h;\theta) = \quad & kE_{P_n}\left[\log \frac{kP_n(w)}{kP_n(w) + P_{\mathcal{D}}(w|h;\theta)}\right] + \\ & E_{P_{\mathcal{D}}(.|h)}\left[\log \frac{P_{\mathcal{D}}(w|h;\theta)}{kP_n(w) + P_{\mathcal{D}}(w|h;\theta)}\right]. \end{aligned} \quad (13)$$

The gradient of this objective function can be computed as

$$\begin{aligned} & \frac{\partial}{\partial\theta} J_{\mathcal{D}}(h;\theta) = \quad (14) \\ & kE_{P_n}\left[\frac{-P_{\mathcal{D}}(w|h;\theta)}{kP_n(w) + P_{\mathcal{D}}(w|h;\theta)} \frac{\partial}{\partial\theta} \log P_{\mathcal{D}}(w|h;\theta)\right] + \\ & E_{P_{\mathcal{D}}(.|h)}\left[\frac{kP_n(w)}{kP_n(w) + P_{\mathcal{D}}(w|h;\theta)} \frac{\partial}{\partial\theta} \log P_{\mathcal{D}}(w|h;\theta)\right] \end{aligned}$$

Since the conditional distributions for different contexts of both domains share parameters, these distributions can then be learned jointly by optimizing a global NCE objective, which is defined as the combination of weighted per-context NCE objectives in the

two domains,

$$J(\theta) = \sum_{\mathcal{D}\in\{\mathcal{S},\mathcal{T}\}} \sum_{h_\mathcal{D}} P(h_\mathcal{D}) J_\mathcal{D}(h_\mathcal{D};\theta) \quad (15)$$

where $P(h_\mathcal{D})$ is the empirical context probability of $h_\mathcal{D}$ in domain $\mathcal{D}$.

In practice, given an observation word w in context h from the domain $\mathcal{D}$, we generate k noise datapoints $x_1, x_2, \ldots, x_k$ from the unigram noise distribution $P_n(w)$, and consider an approximate objective $\widehat{J}_\mathcal{D}(w,h;\theta)$ such that

$$\widehat{J}_\mathcal{D}(w,h;\theta) = \log P_\mathcal{D}(\mathrm{D}=1|w,h;\theta) + \sum_{i=1}^{k} \log\left(1 - P_\mathcal{D}(\mathrm{D}=1|x_i,h;\theta)\right). \quad (16)$$

Its gradient can be computed as

$$\frac{\partial}{\partial\theta}\widehat{J}_\mathcal{D}(w,h;\theta) = \frac{kP_n(w)}{kP_n(w)+P_\mathcal{D}(w|h;\theta)} \frac{\partial}{\partial\theta}\log P_\mathcal{D}(w|h;\theta) - \sum_{i=1}^{k} \frac{P_\mathcal{D}(x_i|h;\theta)}{kP_n(x_i)+P_\mathcal{D}(x_i|h;\theta)} \frac{\partial}{\partial\theta}\log P_\mathcal{D}(x_i|h;\theta). \quad (17)$$

Based on this, we then use an empirical global NCE objective for gradient computation in each iteration of a gradient ascent training procedure, which can be expressed as a sum of the generated approximate objectives for each context-word pair appeared in all sentences of the two domains, i.e.,

$$\widehat{J}(\theta) = \sum_{i=1}^{n_s}\sum_{j=1}^{|X_i^s|} \widehat{J}_\mathcal{S}(w_{ij}^s, h_{ij}^s;\theta) + \sum_{i=1}^{n_t}\sum_{j=1}^{|X_i^t|} \widehat{J}_\mathcal{T}(w_{ij}^t, h_{ij}^t;\theta). \quad (18)$$

Here w_{ij}^s denotes the jth word of the sentence X_i^s, and h_{ij}^s denotes the context of w_{ij}^s, i.e., its previous (n_c-1) words in the sentence X_i^s; the same for w_{ij}^t and its context h_{ij}^t. The gradient of $\widehat{J}(\theta)$ can be easily obtained by summing over the gradients of each context-word pair objective, which can be computed following Equation (17).

3.3. Feature Augmentation with Distributed Representation

After training the LBLA model, we obtain two feature mapping functions, Φ^s (Eq. 2) in the source domain and Φ^t (Eq. 3) in the target domain. Then for each sentence in the source domain, $w_1, w_2, \ldots, w_T$, we use the feature mapping function Φ^s to map each word to a feature vector as augmenting features. Similarly, we produce a augmenting feature vector for each word in the target sentences using the feature mapping function Φ^t. Finally we combine the labeled sentences from both domains, represented using both the original features and the augmenting features, to train supervised NLP systems such as POS tagging, syntactic chunking and named entity recognition, and apply these systems into the target domain.

4. Experiments

We conducted experiments to evaluate the proposed LBLA model based domain adaptation technique on three NLP tasks: POS tagging, syntactic chunking and named entity recognition. In this section, we report the experimental results.

For each task, we compared the proposed LBLA method with the following three baseline methods and four domain adaptation methods: (1) **SRCONLY**, a baseline that conducts training only on the labeled source data; (2) **TGTONLY**, a baseline that conducts training only on the labeled target data; (3) **ALL**, a baseline that conducts training on the labeled data from both domains; (4) **SCL**, the structural correspondence learning (SCL) domain adaptation technique developed in (Blitzer et al., 2006); (5) **LBL**, the method that uses LBL model to produce distributed representation features as augmenting features for NLP systems; (6) **EA**, the feature augmentation based supervised domain adaptation method developed in (Daumé III, 2007); and (7) **EA++**, the feature augmentation based semi-supervised domain adaptation method developed in (Daumé III et al., 2010).

4.1. Domain Adaptation for POS Tagging

For POS tagging, we used the same experimental setting as given in (Daumé III, 2007; Blitzer et al., 2006). 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 1061 labeled sentences and about 100,000 unlabeled sentences. Among the 1061 labeled bio-medical sentences, we used 561 sentences as test data while keeping the rest 500 sentences as labeled training data from the target domain.

4.1.1. Distributed Representation Learning

We built a vocabulary with all sentences from the source and target domain. In order to reduce the vo-

cabulary size, we mapped lower frequency (0-2) words to a single unique identifier in our vocabulary and mapped sole-digit words into a single unique identifier. On all processed sentences except the 561 biomedical sentences which we will keep as test data, we applied the proposed LBLA model to perform distributed representation learning. There are a few hyperparameters to be set when applying LBLA model. We set the word-embedding sizes for source-specific features, target-specific features and domain-sharing (common) features equally as 100. Thus the total size of a word embedding is 300. We set the context size n_c as 3, which means we only consider the previous two words for each target word. We set the k value for noise-contrastive estimation as 25. We randomly initialized the word embeddings R, the position-dependent context weight matrices $C = \{C_i^d : i \in \{1, 2\}, d \in \{c, s, t\}\}$, and initialized the bias vector $\mathbf{b}$, and the normalization parameters $\{z^s(h),\ z^t(h)\}$ with all zeros. The same hyperparameters and initializations were used for LBL model as well. After augmenting each sentence with the learned representation features, standard supervised POS tagging was performed.

4.1.2. Experimental Results for POS Tagging

For supervised POS tagging, we used the SEARN algorithm, which is used in (Daumé III, 2007) as well. We used 39,832 labeled newswire sentences from the WSJ domain and 500 labeled biomedical sentences from the MEDLINE domain as training data, while the test data contains 561 biomedical sentences with 14,554 tokens. Under this setting, the test results of the comparison methods in term of error rate are reported in Table 1. We can see that the LBLA method apparently outperforms all the other comparison methods on cross-domain POS tagging.

We then conducted further experiments to investigate the performance of each method by varying the number of labeled training sentences from the target domain from 50 to 500. The test results in term of accuracy are plotted in Figure 1. We can see that the *LBLA* method consistently outperforms all the other comparison methods across the range of different number of training sentences from the target domain. To investigate the significance of the improvements the proposed LBLA method achieved over the other methods, we conducted McNemar's significance tests for labeling disagreements (Gillick & Cox, 1989) between the LBLA method and the other comparison methods (except the two basic baselines SRCONLY and TGTONLY), with $p < 0.05$ being significant. We found all the test comparisons between LBLA method and the other methods are significant, as shown in Table 2.

Table 1. Test results of POS tagging in term of error rate.

Methods	Error Rates
SRCONLY	12.02%
TGTONLY	4.15%
ALL	5.43%
SCL	3.90%
LBL	3.58%
EA	3.61%
EA++	3.52%
LBLA	**3.09**%

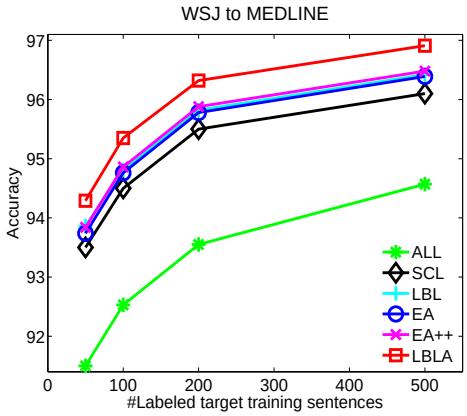

Figure 1. Test results of POS tagging with different number of labeled training sentences from the target domain.

Table 2. Statistical significance (McNemar's) test results in term of p value for cross domain POS tagging.

Null Hypothesis	p-value
LBLA vs. ALL	7.4×10^{-6}
LBLA vs. SCL	3.5×10^{-5}
LBLA vs. LBL	6.9×10^{-3}
LBLA vs. EA	2.1×10^{-4}
LBLA vs. EA++	4.6×10^{-2}

4.2. Domain Adaptation for Syntactic Chunking

For syntactic chunking, we used WSJ as the source domain and Brown corpus data as the target domain. We used the same source domain data as we did in POS tagging experiments. The target domain contains 3 sections (ck01-ck03) of Brown corpus data, with 426 labeled "general fiction" sentences and about 57,000 unlabeled sentences. Labeled sentences from both domains are tagged with syntactic chunking tags in IOB2

Table 3. Test results of syntactic chunking in term of error rate.

METHODS	ERROR RATES
SRCONLY	5.22%
TGTONLY	6.63%
ALL	4.33%
SCL	4.15%
LBL	3.86%
EA	3.97%
EA++	3.82%
LBLA	**3.30**%

format, which is a standard format widely used for syntactic chunking. In IOB2 format, each chunk tag has two parts. The first part denotes the position of the corresponding token in the chunk and the second part represents the chunk type. For example, the chunk tag *B-VP* is used for the first word of a verb phrase.

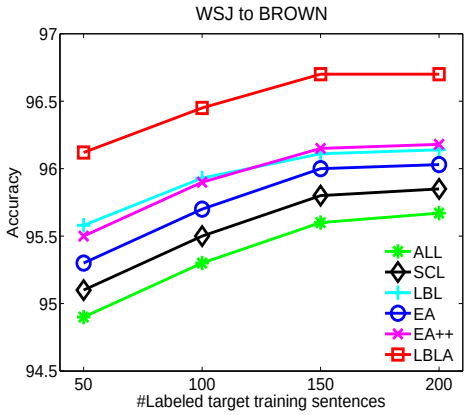

Figure 2. Test results of cross-domain syntactic chunking with different number of labeled training sentences from the target domain.

4.2.1. Distributed Representation Learning

We built a vocabulary with all sentences in the constructed source domain and target domain. We applied the same processing procedures used in the POS tagging experiments. On all processed sentences except 226 "general fiction" sentences from the target domain which we will use as test data, we applied the LBLA and LBL models separately to perform distributed representation learning. We used the same hyperparameter setting and initializations as we did in POS tagging experiments. After augmenting each sentence with the learned representation features, standard supervised syntactic chunking can be performed.

4.2.2. Experimental Results for Syntactic Chunking

We used the same SEARN algorithm for supervised syntactic chunking. We used 40,000 labeled newswire sentences from the source domain and 200 "general fiction" sentences from the target domain as training data, and used 226 "general fiction" sentences from the target domain as test data. In addition to the traditional features, we also extracted POS tag features as inputs. Under this setting, the test results in term of error rate are reported in Table 3, which show the proposed LBLA based cross domain syntactic chunking outperforms all the other methods. We then conducted experiments to investigate the performance of each method by varying the number of labeled training sentences from the target domain between 50 and 200. The test results in term of accuracy are plotted in Figure 2. The proposed method demonstrated consistent advantages over all the other methods. By using McNemar's paired significance tests with $p < 0.05$ being significant, we verified that the proposed LBLA based method significantly outperforms other methods as shown in Table 4.

Table 4. Statistical significance (McNemar's) test results in term of p value for cross domain syntactic chunking.

NULL HYPOTHESIS	P-VALUE
LBLA vs. ALL	7.2×10^{-5}
LBLA vs. SCL	1.9×10^{-4}
LBLA vs. LBL	8.5×10^{-3}
LBLA vs. EA	2.9×10^{-4}
LBLA vs. EA++	3.8×10^{-2}

4.3. Domain Adaptation for Named Entity Recognition

For named entity recognition task, we used the same source data and target data as the syntactic chunking experiments. The labeled data are tagged with named entity chunking tags in IOB2 format. The task is to label each word with one of the named entity tags, which represents the position of the word in the named entity chunk and the type of the named entity. For example, *I-LOC* is used for the remaining words of a phrase that represents a location and *B-PER* is used for the first word of a phrase that represents a person. Words located outside of named entity chunks receive the tag *O*, representing miscellaneous names. We also used the same procedure of distributed presentation learning as we employed in syntactic chunking experiments to produce augmentation features for supervised

Table 5. Test results in term of error rate for cross domain named entity recognition.

METHODS	ERROR RATES
SRCONLY	2.87%
TGTONLY	2.75%
ALL	2.36%
SCL	2.21%
LBL	1.97%
EA	2.06%
EA++	1.93%
LBLA	**1.53**%

named entity recognition.

4.3.1. Experimental Results for Named Entity Recognition

For supervised named entity recognition task, again we used 40,000 labeled newswire sentences from the source domain and 200 labeled "general fiction" sentences from the target domain as training data, and used 226 "general fiction" sentences from the target domain as test data. We used the same SEARN algorithm to perform named entity recognition. In addition to previous feature set, we also extracted syntactic chunking tags as phrase chunking features. The experimental results in term of error rate are reported in Table 5. We can see that the proposed LBLA representation learning based method outperforms all other methods. We then investigated how the number of labeled training sentences from the target domain affects the performance of each comparison method on named entity recognition. The results in term of accuracy are plotted in Figure 3, which show the proposed method clearly outperforms all other methods across the range of experiments. By using McNemar's paired significance test, we verified the improvements achieved by the proposed method over the other methods are mostly significant, as shown in Table 6.

Table 6. Statistical significance (McNemar's) test results in term of p value for cross domain named entity recognition tasks.

NULL HYPOTHESIS	NER
LBLA vs. ALL	6.2×10^{-4}
LBLA vs. SCL	4.3×10^{-3}
LBLA vs. LBL	3.7×10^{-2}
LBLA vs. EA	7.5×10^{-3}
LBLA vs. EA++	8.4×10^{-2}

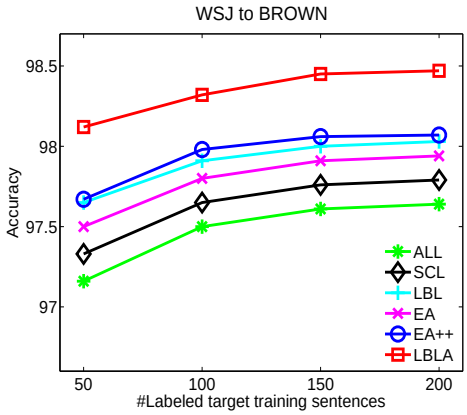

Figure 3. Test results of named entity recognition with different number of labeled training sentences from the target domain.

5. Conclusion

In this paper, we proposed to tackle domain adaptation problems for sequence labeling tasks in NLP by developing a log-bilinear language adaptation (LBLA) model. The LBLA model learns a distributed representation of the words across domains which encodes both generalizable features and domain-specific features. The distributed representation vector for each word can be then used as augmenting features for supervised natural language processing systems. We empirically evaluated the proposed LBLA based domain adaptation method on WSJ and MEDLINE domains for POS tagging systems, and on WSJ and Brown corpora for syntactic chunking and named entity recognition tasks. The results show that LBLA method consistently outperforms all other comparison methods for cross domain sequence labeling tasks.

Acknowledgments

This research was supported by NSF grant IIS-1065397.

References

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

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

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

Bengio, Y. and Senécal, J. Quick training of probabilistic neural nets by importance sampling. In *Proc. of the conference on Artificial Intelligence and Statistics (AISTATS)*, 2003.

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. A neural probabilistic language model. *J. of Machine Learn. Research (JMLR)*, 3:1137–1155, 2003.

Blitzer, J., Weinberger, K., Saul, L., and Pereira, F. Hierarchical distributed representations for statistical language modeling. In *Advances in Neural Information Processing Systems(NIPS)*, 2004.

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

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

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

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

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

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

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

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

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. of Machine Learning Research (JMLR)*, 13:307–361, 2012.

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

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

Jiang, J. and Zhai, C. Instance weighting for domain adaptation in NLP. In *Proc. of the Annual Meeting of the Assoc. of Comput. Linguistics (ACL)*, 2007.

Maas, A. and Ng, A. A probabilistic model for semantic word vectors. In *NIPS Wordshop on Deep Learning and Unsupervised Feature Learning*, 2010.

McClosky, D., Charniak, E., and Johnson, M. Automatic domain adaptation for parsing. In *Proc. of Human Language Technologies: The Annual Conference of the North American Chapter of the Assoc. for Comput. Linguistics (HLT-NAACL)*, 2010.

Mnih, A. and Hinton, G. Three new graphical models for statistical language modelling. In *Proc. of the Inter. Conf. on Machine learning (ICML)*, 2007.

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

Mnih, A. and Teh, Y. A fast and simple algorithm for training neural probabilistic language models. In *Proc. of the International Conference on Machine Learning (ICML)*, 2012.

Socher, R., Lin, C., Ng, A., and Manning, C. Parsing natural scenes and natural language with recursive neural networks. In *Proc. of the International Conference on Machine Learning(ICML)*, 2011.

Turian, J., Ratinov, L., and Bengio, Y. Word representations: a simple and general method for semi-supervised learning. In *Proc. of the Annual Meeting of the Assoc. for Comput. Linguistics (ACL)*, 2010.