Language Models as Representations for Weakly-Supervised NLP Tasks

Fei Huang and **Alexander Yates**
Temple University
Broad St. and Montgomery Ave.
Philadelphia, PA 19122
fei.huang@temple.edu
yates@temple.edu

Arun Ahuja and **Doug Downey**
Northwestern University
2133 Sheridan Road
Evanston, IL 60208
a-ahuja@northwestern.edu
ddowney@eecs.northwestern.edu

Abstract

Finding the right representation for words is critical for building accurate NLP systems when domain-specific labeled data for the task is scarce. This paper investigates language model representations, in which language models trained on unlabeled corpora are used to generate real-valued feature vectors for words. We investigate ngram models and probabilistic graphical models, including a novel lattice-structured Markov Random Field. Experiments indicate that language model representations outperform traditional representations, and that graphical model representations outperform ngram models, especially on sparse and polysemous words.

1 Introduction

NLP systems often rely on hand-crafted, carefully engineered sets of features to achieve strong performance. Thus, a part-of-speech (POS) tagger would traditionally use a feature like, "the previous token is `the`" to help classify a given token as a noun or adjective. For supervised NLP tasks with sufficient domain-specific training data, these traditional features yield state-of-the-art results. However, NLP systems are increasingly being applied to texts like the Web, scientific domains, and personal communications like emails, all of which have very different characteristics from traditional training corpora. Collecting labeled training data for each new target domain is typically prohibitively expensive. We investigate representations that can be applied when domain-specific labeled training data is scarce.

An increasing body of theoretical and empirical evidence suggests that traditional, manually-crafted features limit systems' performance in this setting for two reasons. First, feature *sparsity* prevents systems from generalizing accurately to words and features not seen during training. Because word frequencies are Zipf distributed, this often means that there is little relevant training data for a substantial fraction of parameters (Bikel, 2004), especially in new domains (Huang and Yates, 2009). For example, word-type features form the backbone of most POS-tagging systems, but types like "gene" and "pathway" show up frequently in biomedical literature, and rarely in newswire text. Thus, a classifier trained on newswire data and tested on biomedical data will have seen few training examples related to sentences with features "gene" and "pathway" (Ben-David et al., 2009; Blitzer et al., 2006).

Further, because words are *polysemous*, word-type features prevent systems from generalizing to situations in which words have different meanings. For instance, the word type "signaling" appears primarily as a present participle (VBG) in Wall Street Journal (WSJ) text, as in, "Interest rates rose, signaling that . . . " (Marcus et al., 1993). In biomedical text, however, "signaling" appears primarily in the phrase "signaling pathway," where it is considered a noun (NN) (PennBioIE, 2005); this phrase never appears in the WSJ portion of the Penn Treebank (Huang and Yates, 2010a).

Our response to these problems with traditional NLP representations is to seek new representations that allow systems to generalize more accurately to previously unseen examples. Our approach depends on the well-known *distributional hypothesis*, which states that a word's meaning is identified with the contexts in which it appears (Harris, 1954; Hindle, 1990). Our goal is to develop probabilistic lan-

Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pages 125–134,
Portland, Oregon, USA, 23–24 June 2011.

guage models that describe the contexts of individual words accurately. We then construct *representations*, or mappings from word tokens and types to real-valued vectors, from these language models. Since the language models are designed to model words' contexts, the features they produce can be used to combat problems with polysemy. And by careful design of the language models, we can limit the number of features that they produce, controlling how sparse those features are in training data.

In this paper, we analyze the performance of language-model-based representations on tasks where domain-specific training data is scarce. Our contributions are as follows:

1. We introduce a novel factorial graphical model representation, a Partial-Lattice Markov Random Field (PL-MRF), which is a tractable variation of a Factorial Hidden Markov Model (HMM) for language modeling.
2. In experiments on POS tagging in a domain adaptation setting and on weakly-supervised information extraction (IE), we quantify the performance of representations derived from language models. We show that graphical models outperform ngram representations. The PL-MRF representation achieves a state-of-the-art 93.8% accuracy on the POS tagging task, while the HMM representation improves over the ngram model by 10% on the IE task.
3. We analyze how the performance of the different representations varies due to the fundamental challenges of sparsity and polysemy.

The next section discusses previous work. Sections 3 and 4 present the existing representations we investigate and the new PL-MRF, respectively. Sections 5 and 6 describe our two tasks and the results of using our representations on each of them. Section 7 concludes.

2 Previous Work

There is a long tradition of NLP research on representations, mostly falling into one of four categories: 1) vector space models of meaning based on document-level lexical cooccurrence statistics (Salton and McGill, 1983; Turney and Pantel, 2010; Sahlgren, 2006); 2) dimensionality reduction techniques for vector space models (Deerwester et al., 1990; Honkela, 1997; Kaski, 1998; Sahlgren, 2005; Blei et al., 2003; Väyrynen et al., 2007); 3) using clusters that are induced from distributional similarity (Brown et al., 1992; Pereira et al., 1993; Martin et al., 1998) as non-sparse features (Lin and Wu, 2009; Candito and Crabbe, 2009; Koo et al., 2008; Zhao et al., 2009); 4) and recently, language models (Bengio, 2008; Mnih and Hinton, 2009) as representations (Weston et al., 2008; Collobert and Weston, 2008; Bengio et al., 2009), some of which have already yielded state of the art performance on domain adaptation tasks (Huang and Yates, 2009; Huang and Yates, 2010a; Huang and Yates, 2010b; Turian et al., 2010) and IE (Ahuja and Downey, 2010; Downey et al., 2007b). In contrast to this previous work, we develop a novel Partial Lattice MRF language model that incorporates a factorial representation of latent states, and demonstrate that it outperforms the previous state-of-the-art in POS tagging in a domain adaptation setting. We also analyze the novel PL-MRF representation on an IE task, and several representations along the key dimensions of sparsity and polysemy.

Most previous work on domain adaptation has focused on the case where some labeled data is available in both the source and target domains (Daumé III, 2007; Jiang and Zhai, 2007; Daumé III and Marcu, 2006; Finkel and Manning, 2009; Dredze et al., 2010; Dredze and Crammer, 2008). Learning bounds are known (Blitzer et al., 2007; Mansour et al., 2009). Daumé III *et al.* (2010) use semi-supervised learning to incorporate labeled and unlabeled data from the target domain. In contrast, we investigate a domain adaptation setting where no labeled data is available for the target domain.

3 Representations

A *representation* is a set of features that describe instances for a classifier. Formally, let $\mathcal{X}$ be an instance set, and let $\mathcal{Z}$ be the set of labels for a classification task. A representation is a function $R : \mathcal{X} \rightarrow \mathcal{Y}$ for some suitable feature space $\mathcal{Y}$ (such as $\mathbb{R}^d$). We refer to dimensions of $\mathcal{Y}$ as *features*, and for an instance $x \in \mathcal{X}$ we refer to values for particular dimensions of $R(x)$ as features of x.

3.1 Traditional POS-Tagging Representations

As a baseline for POS tagging experiments and an example of our terminology, we describe a representation used in traditional supervised POS taggers. The instance set $\mathcal{X}$ is the set of English sentences, and $\mathcal{Z}$ is the set of POS tag sequences. A traditional representation TRAD-R maps a sentence $\mathbf{x} \in \mathcal{X}$ to a sequence of boolean-valued vectors, one vector per

Representation	Feature
TRAD-R	$\forall_w \mathbf{1}[x_i = w]$ $\forall_{s \in \text{Suffixes}} \mathbf{1}[x_i \text{ ends with } s]$ $\mathbf{1}[x_i \text{ contains a digit}]$
NGRAM-R	$\forall_{\mathbf{w}',\mathbf{w}''} P(\mathbf{w}' w \mathbf{w}'')/P(w)$
HMM-TOKEN-R	$\forall_k \mathbf{1}[y_i* = k]$
HMM-TYPE-R	$\forall_k P(y = k \mid x = w)$
I-HMM-TOKEN-R	$\forall_{j,k} \mathbf{1}[y_{i,j}* = k]$
BROWN-TOKEN-R	$\forall_{j \in \{-2,-1,0,1,2\}}$ $\forall_{p \in \{4,6,10,20\}}$ prefix(y_{i+j}, p)
BROWN-TYPE-R	$\forall_p$ prefix(y, p)
LATTICE-TOKEN-R	$\forall_{j,k} \mathbf{1}[y_{i,j}* = k]$
LATTICE-TYPE-R	$\forall_{\mathbf{k}} P(\mathbf{y} = \mathbf{k} \mid x = w)$

Table 1: Summary of features provided by our representations. $\forall_a \mathbf{1}[g(a)]$ represents a set of boolean features, one for each value of a, where the feature is true iff $g(a)$ is true. x_i represents a token at position i in sentence $\mathbf{x}$, w represents a word type, Suffixes = {-ing,-ogy,-ed,-s,-ly,-ion,-tion,-ity}, k (and $\mathbf{k}$) represents a value for a latent state (set of latent states) in a latent-variable model, $\mathbf{y}*$ represents the optimal setting of latent states $\mathbf{y}$ for $\mathbf{x}$, y_i is the latent variable for x_i, and $y_{i,j}$ is the latent variable for x_i at layer j. prefix(y,p) is the p-length prefix of the Brown cluster y.

word x_i in the sentence. Dimensions for each latent vector include indicators for the word type of x_i and various orthographic features. Table 1 presents the full list of features in TRAD-R. Since our IE task classifies word types rather than tokens, this baseline is not appropriate for that task. Below, we describe how we can learn representations R by using a variety of language models, for use in both our IE and POS tagging tasks. All representations for POS tagging inherit the features from TRAD-R; all representations for IE do not.

3.2 Ngram Representations

N-gram representations model a word type w in terms of the n-gram contexts in which w appears in a corpus. Specifically, for word w we generate the vector $P(\mathbf{w}' w \mathbf{w}'')/P(w)$, the conditional probability of observing the word sequence $\mathbf{w}'$ to the left and $\mathbf{w}''$ to the right of w. The experimental section describes the particular corpora and language modeling methods used for estimating probabilities.

3.3 HMM-based Representations

In previous work, we have implemented several representations based on HMMs (Rabiner, 1989), which we used for both POS tagging (Huang and Yates, 2009) and IE (Downey et al., 2007b). An HMM is a generative probabilistic model that generates each word x_i in the corpus conditioned on a latent variable y_i. Each y_i in the model takes on integral values from 1 to K, and each one is generated by the latent variable for the preceding word, y_{i-1}. The joint distribution for a corpus $\mathbf{x} = (x_1, \ldots, x_N)$ and a set of state vectors $\mathbf{y} = (y_1, \ldots, y_N)$ is given by: $P(\mathbf{x}, \mathbf{y}) = \prod_i P(x_i|y_i)P(y_i|y_{i-1})$. Using Expectation-Maximization (EM) (Dempster et al., 1977), it is possible to estimate the distributions for $P(x_i|y_i)$ and $P(y_i|y_{i-1})$ from unlabeled data.

We construct two different representations from HMMs, one for POS tagging and one for IE. For POS tagging, we use the Viterbi algorithm to produce the optimal setting $\mathbf{y}*$ of the latent states for a given sentence $\mathbf{x}$, or $\mathbf{y}* = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y})$. We use the value of y_i* as a new feature for x_i that represents a cluster of distributionally-similar words. For IE, we require features for word types w, rather than tokens x_i. We use the K-dimensional vector that represents the distribution $P(y|x = w)$ as the feature vector for word type w. This set of features represents a "soft clustering" of w into K different clusters. We refer to these representations as HMM-TOKEN-R and HMM-TYPE-R, respectively.

Because HMM-based representations offer a small number of discrete states as features, they have a much greater potential to combat feature sparsity than do ngram models. Furthermore, for token-based representations, these models can potentially handle polysemy better than ngram language models by providing different features in different contexts.

We also compare against a variation of the HMM from our previous work (Huang and Yates, 2010a), henceforth HY10. This model independently trains M separate HMM models on the same corpus, initializing each one randomly. We can then use the Viterbi-optimal decoded latent state of each independent HMM model as a separate feature for a token. We refer to this language model as an I-HMM, and the representation as I-HMM-TOKEN-R.

Finally, we compare against Brown clusters (Brown et al., 1992) as learned features. Although not traditionally described as such, Brown clustering involves constructing an HMM model in which

each type is restricted to having exactly one latent state that may generate it. Brown *et al.* describe a greedy agglomerative clustering algorithm for training this model on unlabeled text. Following Turian *et al.* (2010), we use Percy Liang's implementation of this algorithm for our comparison, and we test runs with 100, 320, and 1000 clusters. We use features from these clusters identical to Turian *et al.*'s.[1] Turian *et al.* have shown that Brown clusters match or exceed the performance of neural network-based language models in domain adaptation experiments for named-entity recognition, as well as in-domain experiments for NER and chunking.

4 A Novel Lattice Language Model Representation

Our final language model is a novel latent-variable language model with rich latent structure, shown in Figure 1. The model contains a lattice of $M \times N$ latent states, where N is the number of words in a sentence and M is the number of layers in the model. We can justify the choice of this model from a linguistic perspective as a way to capture the multi-dimensional nature of words. Linguists have long argued that words have many different features in a high dimensional space: they can be separately described by part of speech, gender, number, case, person, tense, voice, aspect, mass vs. count, and a host of semantic categories (agency, animate vs. inanimate, physical vs. abstract, etc.), to name a few (Sag et al., 2003). Our model seeks to capture a multi-dimensional representation of words by creating a separate layer of latent variables for each dimension. The values of the M layers of latent variables for a single word can be used as M distinct features in our representation. The I-HMM attempts to model the same intuition, but unlike a lattice model the I-HMM layers are entirely independent, and as a result there is no mechanism to enforce that the layers model different dimensions. Duh (2005) previously used a 2-layer lattice for tagging and chunking, but in a supervised setting rather than for representation learning.

Let $Cliq(\mathbf{x}, \mathbf{y})$ represent the set of all maximal cliques in the graph of the MRF model for $\mathbf{x}$ and $\mathbf{y}$.

[1]Percy Liang's implementation is available at *http://metaoptimize.com/projects/wordreprs/*. Turian *et al.* also tested a run with 3200 clusters in their experiments, which we have been training for months, but which has not finished in time for publication.

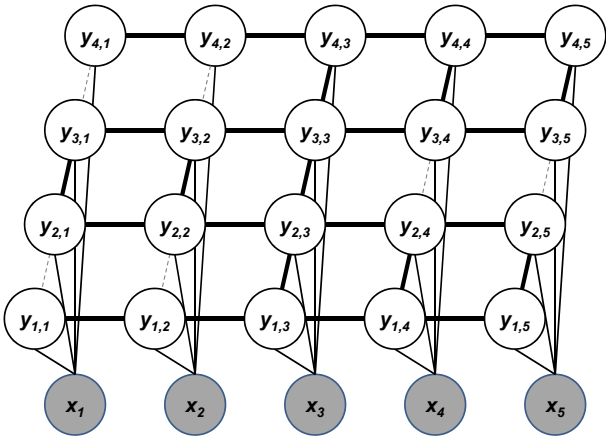

Figure 1: The Partial Lattice MRF (PL-MRF) Model for a 5-word sentence and a 4-layer lattice. Dashed gray edges are part of a full lattice, but not the PL-MRF.

Expressing the lattice model in log-linear form, we can write the marginal probability $P(\mathbf{x})$ of a given sentence $\mathbf{x}$ as:

$$\sum_{\mathbf{y}} \frac{\prod_{c \in Cliq(\mathbf{x},\mathbf{y})} \text{score}(c, \mathbf{x}, \mathbf{y})}{\sum_{\mathbf{x}',\mathbf{y}'} \prod_{c \in Cliq(\mathbf{x}',\mathbf{y}')} \text{score}(c, \mathbf{x}', \mathbf{y}')}$$

where $\text{score}(c, \mathbf{x}, \mathbf{y}) = \exp(\theta_c \cdot \mathbf{f}_c(\mathbf{x}_c, \mathbf{y}_c))$. Our model includes parameters for transitions between two adjacent latent variables on layer j: $\theta^{trans}_{i,s,i+1,s',j}$ for $y_{i,j} = s$ and $y_{i+1,j} = s'$. It also includes observation parameters for latent variables and tokens, as well as for pairs of adjacent latent variables in different layers and their tokens: $\theta^{obs}_{i,j,s,w}$ and $\theta^{obs}_{i,j,s,j+1,s',w}$ for $y_{i,j} = s$, $y_{i,j+1} = s'$, and $x_i = w$.

Computationally, the lattice MRF is preferable to a naïve Factorial HMM (Ghahramani and Jordan, 1997) representation, which would require $O(2^M)$ parameters for an M-layer model. However, exact training and inference in supervised settings are still intractable for this model (Sutton et al., 2007), and thus it has not yet been explored as a language model, which requires even more difficult, unsupervised training. Training is intractable in part because of the difficulty in enumerating and summing over the exponentially-many configurations $\mathbf{y}$ for a given $\mathbf{x}$. We address this difficulty in two ways: by modifying the model, and by modifying the training procedure.

4.1 Partial Lattice MRF

Instead of the full lattice model, we construct a *Partial Lattice MRF* (PL-MRF) model by deleting

certain edges between latent layers of the model (dashed gray edges in Figure 1). Let $c = \lfloor \frac{N}{2} \rfloor$, where N is the length of the sentence. If $i < c$ and j is odd, or if j is even and $i > c$, we delete edges between $y_{i,j}$ and $y_{i,j+1}$. The same lattice of nodes remains, but fewer edges and paths. A central "trunk" at $i = c$ connects all layers of the lattice, and branches from this trunk connect either to the branches in the layer above or the layer below (but not both). The result is a model that retains most[2] of the edges of the full model. Additionally, the pruned model makes the branches conditionally independent from one another, except through the trunk. For instance, the right branch at layers 1 and 2 in Figure 1 ($y_{1,4}, y_{1,5}, y_{2,4}$, and $y_{2,5}$) are disconnected from the right branch at layers 3 and 4 ($y_{3,4}, y_{3,5}, y_{4,4}$, and $y_{4,5}$), except through the trunk and the observed nodes. As a result, excluding the observed nodes, this model has a low *tree-width* of 2 (excluding observed nodes), and a variety of efficient dynamic programming and message-passing algorithms for training and inference can be readily applied (Bodlaender, 1988).[3] Our inference algorithm passes information from the branches inwards to the trunk, and then upward along the trunk, in time $O(K^4MN)$.

As with our HMM models, we create two representations from PL-MRFs, one for tokens and one for types. For tokens, we decode the model to compute $\mathbf{y}*$, the matrix of optimal latent state values for sentence $\mathbf{x}$. For each layer j and and each possible latent state value k, we add a boolean feature for token x_i that is true iff $\mathbf{y}*_{i,j} = k$. For types, we compute distributions over the latent state space. Let $\mathbf{y}$ be the column vector of latent variables for word x. For each possible configuration of values $\mathbf{k}$ of the latent variables $\mathbf{y}$, we add a real-valued features for x given by $P(\mathbf{y} = \mathbf{k}|x = w)$. We refer to these two representations as LATTICE-TOKEN-R and LATTICE-TYPE-R, respectively.

4.2 Parameter Estimation

We train the PL-MRF using contrastive estimation, which iteratively optimizes the following objective function on a corpus $\mathbf{X}$:

$$\sum_{\mathbf{x} \in \mathbf{X}} \log \frac{\sum_{\mathbf{y}} \prod_{c \in Cliq(\mathbf{x},\mathbf{y})} \text{score}(c, \mathbf{x}, \mathbf{y})}{\sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}), \mathbf{y}'} \prod_{c \in Cliq(\mathbf{x}',\mathbf{y}')} \text{score}(c, \mathbf{x}', \mathbf{y}')}$$

where $\mathcal{N}(\mathbf{x})$, the neighborhood of $\mathbf{x}$, indicates a set of perturbed variations of the original sentence $\mathbf{x}$. Contrastive estimation seeks to move probability mass away from the perturbed neighborhood sentences and onto the original sentence. We use a neighborhood function that includes all sentences which can be obtained from the original sentence by swapping the order of a consecutive pair of words. Training uses gradient descent over this non-convex objective function with a standard software package (Liu and Nocedal, 1989) and converges to a local maximum (Smith and Eisner, 2005).

For tractability, we modify the training procedure to train the PL-MRF one layer at a time. Let θ_i represent the set of parameters relating to features of layer i, and let $\theta_{\neg i}$ represent all other parameters. We fix $\theta_{\neg 0} = \mathbf{0}$, and optimize θ_0 using contrastive estimation. After convergence, we fix $\theta_{\neg 1}$, and optimize θ_1, and so on. We use a convergence threshold of 10^{-6}, and each layer typically converges in under 100 iterations.

5 Domain Adaptation for a POS Tagger

We evaluate the representations described above on a POS tagging task in a domain adaptation setting.

5.1 Experimental Setup

We use the same experimental setup as in HY10: the Penn Treebank (Marcus et al., 1993) Wall Street Journal portion for our labeled training data; 561 MEDLINE sentences (9576 types, 14554 tokens, 23% OOV tokens) 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 a MEDLINE corpus of 71,306 unlabeled sentences to train our language models. The two texts come from two very different domains, making this data a tough test for domain adaptation.

We use an open source Conditional Random Field (CRF) (Lafferty et al., 2001) software package[4] designed by Sunita Sarawagi and William W. Cohen to implement our supervised models. Let $\mathbf{X}$ be a training corpus, $\mathbf{Z}$ the corresponding labels, and R a representation function. For each token x_i in $\mathbf{X}$, we include a parameter in our CRF model for all features $R(x_i)$ and all possible labels in $\mathbf{Z}$. Furthermore, we include transition parameters for pairs of consecutive labels z_i, z_{i+1}.

[2] As $M, N \to \infty$, 5 out of every 6 edges are kept.

[3] *c.f.* a tree-width of min(M,N) for the unpruned model

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

For representations, we tested TRAD-R, NGRAM-R, HMM-TOKEN-R, I-HMM-TOKEN-R (between 2 and 8 layers), and LATTICE-TOKEN-R (8, 12, 16, and 20 layers). Following HY10, each latent node in the I-HMMs have 80 possible values, creating $80^8 \approx 10^{15}$ possible configurations of the 8-layer I-HMM for a single word. Each node in the PL-MRF is binary, creating a much smaller number ($2^{20} \approx 10^6$) of possible configurations for each word in a 20-layer representation. NGRAM-R was trained using an unsmoothed trigram model on the Web 1Tgram corpus. To keep the feature set manageable, we included the top 500 most common ngrams for each word type, and then used mutual information on the training data to select the top 10,000 most relevant ngram features for all word types. We incorporated ngram features as binary values indicating whether x_i appeared with the ngram or not. We also report on the performance of Brown clusters and Blitzer *et al.*'s Structural Correspondence Learning (SCL) (2006) technique, which uses manually-selected "pivot" words (like "of", "the") to learn domain-independent features. Finally, we compare against the self-training CRF technique from HY10.

5.2 Results and Discussion

For each representation, we measured the accuracy of the POS tagger on the biomedical test text. Table 2 shows the results for the best variation of each kind of model — 20 layers for the PL-MRF, 7 layers for the I-HMM, and 1000 clusters for the Brown clustering. All language model representations significantly outperform the SCL model and the TRAD-R baseline. The novel PL-MRF model outperforms the previous state of the art, the I-HMM model, and much of the performance increase comes from a 11.3% relative reduction in error on words that appear in biomedical texts but not in newswire texts. Both graphical model representations significantly outperform the ngram model, which is trained on far more text. For comparison, our best model, the PL-MRF, achieved a 96.8% in-domain accuracy on sections 22-24 of the Penn Treebank, about 0.5% shy of a state-of-the-art in-domain system (Shen et al., 2007) with more sophisticated supervised learning.

We expected that language model representations perform well in part because they provide meaningful features for sparse and polysemous words. To test this, we selected 109 polysemous word types from our test data, along with 296 non-polysemous word types, chosen based on POS tags and manual inspection. We further define sparse word types as those that appear 5 times or fewer in all of our unlabeled data, and non-sparse word types as those that appear at least 50 times in our unlabeled data. Table 3 shows results on these subsets of the data.

model	% error	OOV % error
TRAD-R	11.7	32.7
TRAD-R+self-training	11.5	29.6
SCL	11.1	-
BROWN-TOKEN-R	10.8	25.4
HMM-TOKEN-R	9.5	24.8
NGRAM-R	6.9	24.4
I-HMM-TOKEN-R	6.7	24
LATTICE-TOKEN-R	**6.2**	**21.3**
SCL+500bio	3.9	-

Table 2: PL-MRF representations reduce error by 7.5% relative to the previous state-of-the-art I-HMM, and approach within 2.3% absolute error a SCL+500bio model with access to 500 labeled sentences from the target domain. 1.8% of the tags in the test set are new tags that do not occur in the WSJ training data, so an error rate of 3.9+1.8 = 5.7% error is a reasonable bound for the best possible performance of a model that has seen no examples from the target domain.

As expected, all of our language models outperform the baseline by a larger margin on polysemous words than on non-polysemous words. The margin between graphical model representations and the ngram model also increases on polysemous words, presumably because the Viterbi decoding of these models takes into account the tokens in the surrounding sentence. The same behavior is evident for sparsity: all of the language model representations outperform the baseline by a larger margin on sparse words than not-sparse words, and all of the graphical models perform better relative to the ngram model on sparse words as well. Thus representations based on graphical models address two key issues in building representations for POS tagging.

6 Information Extraction Experiments

In this section, we evaluate our learned representations on a different task that investigates the ability of each representation to capture semantic, rather than syntactic, information. Specifically, we inves-

	POS Tagging				Information Extraction			
	polys.	not polys.	sparse	not sparse	polys.	not polys.	sparse	not sparse
tokens/types	159	4321	463	12194	222	210	266	166
categories	-	-	-	-	12	4	13	3
TRAD-R	59.5	78.5	52.5	89.6	-	-	-	-
Ngram	68.2	85.3	61.8	94.0	0.07	0.17	0.06	0.25
HMM	67.9	83.4	60.2	91.6	**0.14**	**0.26**	**0.15**	**0.32**
(-Ngram)	(-0.3)	(-1.9)	(-1.6)	(-2.4)	(**+0.07**)	(**+0.09**)	(**+0.09**)	(**+0.07**)
I-HMM	**75.6**	85.2	62.9	94.5	-	-	-	-
(-Ngram)	(**+7.4**)	(-0.1)	(+1.1)	(+0.5)	-	-	-	-
PL-MRF	70.5	**86.9**	**65.2**	**94.6**	0.09	0.15	0.1	0.19
(-Ngram)	(+2.3)	(**+1.6**)	(**+3.4**)	(**+0.6**)	(+0.02)	(-0.02)	(+0.04)	(-0.06)

Table 3: Graphical models consistently outperform ngram models by a larger margin on sparse words than not-sparse words. On polysemous words, the difference between graphical model performance and ngram performance grows for POS tagging, where the context surrounding polysemous words is available to the language model, but not for information extraction. For tagging, we show number of tokens and accuracies. For IE, we show number of types, categories, and AUCs.

tigate a *set-expansion* task in which we're given a corpus and a few "seed" noun phrases from a semantic category (e.g. Superheroes), and our goal is to identify other examples of the category in the corpus. This is a *weakly-supervised* task because we are given only a handful of examples of the category, rather than a large sample of positively and negatively labeled training examples.

Existing set-expansion techniques utilize the distributional hypothesis: candidate noun phrases for a given semantic class are ranked based on how similar their contextual distributions are to those of the seeds. Here, we measure how performance on the set-expansion task varies when we employ different representations for the contextual distributions.

6.1 Methods

The set-expansion task we address is formalized as follows: given a corpus, a set of seeds from some semantic category C, and a separate set of candidate phrases P, output a ranking of the phrases in P in decreasing order of likelihood of membership in C.

For any given representation R, the set-expansion algorithm we investigate is straightforward: we create a prototypical "seed representation vector" equal to the mean of the representation vectors for each of the seeds. Then, we rank candidate phrases in increasing order of the distance between the candidate phrase representation and the seed representation vector. As a measure of distance between representations, we compute the average of five standard distance measures, including KL and Jensen-Shannon divergence, and cosine, Euclidean, and L1 distance. In experiments, we found that improving upon this simple averaging was not easy—in fact, tuning a weighted average of the distance measures for each representation did not improve results significantly on held-out data.

Because set expansion is performed at the level of word types rather than tokens, it requires type-based representations. We compare HMM-TYPE-R, NGRAM-R, LATTICE-TYPE-R, and BROWN-TYPE-R in this experiment. We used a 25-state HMM, and the same PL-MRF as in the previous section. Following previous set-expansion experiments with n-grams (Ahuja and Downey, 2010), we employ a trigram model with Kneser-Ney smoothing for NGRAM-R. For Brown clusters, instead of distance metrics like KL divergence (which assume distributions), we rank extractions by the number of matches between a word's BROWN-TYPE-R features and seed features.

6.2 Data Sets

We utilized a set of approximately 100,000 sentences of Web text, joining multi-word named entities in the corpus into single tokens using the Lex algorithm (Downey et al., 2007a). This process enables each named entity (the focus of the set-expansion experiments) to be treated as a single token, with a single representation vector for comparison. We developed all word type representations

model	AUC
HMM-TYPE-R	**0.18**
BROWN-TYPE-R	0.16
LATTICE-TYPE-R	0.11
NGRAM-R	0.10
Random baseline	0.10

Table 4: HMM-TYPE-R outperforms the other methods, improving performance by 12.5% over Brown clusters, and by 80% over the traditional NGRAM-R.

using this corpus.

To obtain examples of multiple semantic categories, we utilized selected Wikipedia "listOf" pages from (Pantel et al., 2009) and augmented these with our own manually defined categories, such that each list contained at least ten distinct examples occurring in our corpus. In all, we had 432 examples across 16 distinct categories such as Countries, Greek Islands, and Police TV Dramas.

6.3 Results

For each semantic category, we tested five different random selections of five seed examples, treating the unselected members of the category as positive examples, and all other candidate phrases as negative examples. We evaluate using the area under the precision-recall curve (AUC) metric.

The results are shown in Table 4. All representations improve performance over a random baseline, equal to the average AUC over five random orderings for each category, and the graphical models outperform the ngram representation. HMM-TYPE-R performs the best overall, and Brown clustering with 1000 clusters is comparable (320 and 100 cluster perform slightly worse).

As with POS tagging, we expect that language model representations improve performance on the IE task by providing informative features for sparse word types. However, because the IE task classifies word types rather than tokens, we expect the representations to provide less benefit for polysemous word types. To test these hypotheses, we measured how IE performance changed in sparse or polysemous settings. We identified polysemous categories as those for which fewer than 90% of the category members had the category as a clear dominant sense (estimated manually); other categories were considered non-polysemous. Categories whose members had a median number of occurrences in the corpus less than 30 were deemed sparse, and others non-sparse. IE performance on these subsets of the data are shown in Table 3. Both graphical model representations outperform the ngram representation more on sparse words, as expected. For polysemy, the picture is mixed: the PL-MRF outperform n-grams on polysemous categories, whereas HMM's performance advantage over n-grams decreases.

One surprise on the IE task is that the LATTICE-TYPE-R performs significantly less well than the HMM-TYPE-R, whereas the reverse is true on POS tagging. We suspect that the difference is due to the issue of classifying types vs. tokens. Because of their more complex structure, PL-MRFs tend to depend more on transition parameters than do HMMs. Furthermore, our decision to train the PL-MRFs using contrastive estimation with a neighborhood that swaps consecutive pairs of words also tends to emphasize transition parameters. As a result, we believe the posterior distribution over latent states given a word type is more informative in our HMM model than the PL-MRF model. We measured the entropy of these distributions for the two models, and found that $H(P_{\text{PL-MRF}}(\mathbf{y}|x = w)) = 9.95$ bits, compared with $H(P_{\text{HMM}}(\mathbf{y}|x = w)) = 2.74$ bits, which supports the hypothesis that the drop in the PL-MRF's performance on IE is due to its dependence on transition parameters. Further experiments are warranted to investigate this issue.

7 Conclusion and Future Work

Our investigation into language models as representations shows that graphical models can be used to combat polysemy and, especially, sparsity in representations for weakly-supervised classifiers. Our novel factorial graphical model yields a state-of-the-art POS tagger for domain adaptation, and HMMs improve significantly over all other representations in an information extraction task. Important directions for future research include models for handling polysemy in IE, and richer language models that incorporate more linguistic intuitions about how words interact with their contexts.

Acknowledgments

This research was supported in part by NSF grant IIS-1065397 and a Microsoft New Faculty Fellowship.

References

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

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. 2009. A theory of learning from different domains. *Machine Learning*, (to appear).

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *International Conference on Machine Learning (ICML)*.

Yoshua Bengio. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

Daniel M. Bikel. 2004. Intricacies of Collins Parsing Model. *Computational Linguistics*, 30(4):479–511.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January.

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*.

Hans L. Bodlaender. 1988. Dynamic programming on graphs with bounded treewidth. In *Proc. 15th International Colloquium on Automata, Languages and Programming*, pages 105–118.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, pages 467–479.

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

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

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

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

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

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

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

D. Downey, M. Broadhead, and O. Etzioni. 2007a. Locating complex named entities in web text. In *Procs. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*.

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

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of EMNLP*, pages 689–697.

Mark Dredze, Alex Kulesza, and Koby Crammer. 2010. Multi-domain learning by confidence weighted parameter combination. *Machine Learning*, 79.

Kevin Duh. 2005. Jointly labeling multiple sequences: A Factorial HMM approach. In *43rd Annual Meeting of the Assoc. for Computational Linguistics (ACL 2005), Student Research Workshop*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of HLT-NAACL*, pages 602–610.

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

Z. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

D. Hindle. 1990. Noun classification from predicage-argument structures. In *ACL*.

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

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

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

Fei Huang and Alexander Yates. 2010b. Open-domain semantic role labeling by modeling word spans. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

S. Kaski. 1998. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *IJCNN*, pages 413–418.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 595–603.

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*.

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

D.C. Liu and J. Nocedal. 1989. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

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

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.

S. Martin, J. Liermann, and H. Ney. 1998. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24:19–37.

A. Mnih and G. E. Hinton. 2009. A scalable hierarchical distributed language model. In *Neural Information Processing Systems (NIPS)*, pages 1081–1088.

P. Pantel, E. Crestan, A. Borkovsky, A. M. Popescu, and V. Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proc. of EMNLP*.

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

F. Pereira, N. Tishby, and L. Lee. 1993. Distributional clustering of English words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 183–190.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Synactic Theory: A Formal Introduction*. CSLI, Stanford, CA, second edition.

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

M. Sahlgren. 2006. *The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Stockholm University.

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

L. Shen, G. Satta, and A. Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 760–767.

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.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723.

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

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

J. J. Väyrynen, T. Honkela, and L. Lindqvist. 2007. Towards explicit semantic features using independent component analysis. In *Proceedings of the Workshop Semantic Content Acquisition and Representation (SCAR)*.

Jason Weston, Frederic Ratle, and Ronan Collobert. 2008. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*.

Hai Zhao, Wenliang Chen, Chunyu Kit, and Guodong Zhou. 2009. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *CoNLL 2009 Shared Task*.