Modeling Latent-Dynamic in Shallow Parsing: A Latent Conditional Model with Improved Inference

Xu Sun† **Louis-Philippe Morency**‡ **Daisuke Okanohara**† **Jun'ichi Tsujii**†§
†Department of Computer Science, The University of Tokyo, Hongo 7-3-1, Tokyo, Japan
‡USC Institute for Creative Technologies, 13274 Fiji Way, Marina del Rey, USA
§School of Computer Science, The University of Manchester, 131 Princess St, Manchester, UK
†{sunxu, hillbig, tsujii}@is.s.u-tokyo.ac.jp ‡morency@ict.usc.edu

Abstract

Shallow parsing is one of many NLP tasks that can be reduced to a sequence labeling problem. In this paper we show that the *latent-dynamics* (i.e., hidden substructure of shallow phrases) constitutes a problem in shallow parsing, and we show that modeling this intermediate structure is useful. By analyzing the automatically learned hidden states, we show how the latent conditional model explicitly learn latent-dynamics. We propose in this paper the Best Label Path (BLP) inference algorithm, which is able to produce the most probable label sequence on latent conditional models. It outperforms two existing inference algorithms. With the BLP inference, the LDCRF model significantly outperforms CRF models on word features, and achieves comparable performance of the most successful shallow parsers on the CoNLL data when further using part-of-speech features.

1 Introduction

Shallow parsing identifies the non-recursive cores of various phrase types in text. The paradigmatic shallow parsing problem is noun phrase chunking, in which the non-recursive cores of noun phrases, called base NPs, are identified. As the representative problem in shallow parsing, noun phrase chunking has received much attention, with the development of standard evaluation datasets and with extensive comparisons among methods (McDonald 2005; Sha & Pereira 2003; Kudo & Matsumoto 2001).

Syntactic contexts often have a complex underlying structure. Chunk labels are usually far too general to fully encapsulate the syntactic behavior of word sequences. In practice, and given the limited data, the relationship between specific words and their syntactic contexts may be best modeled at a level finer than chunk tags but coarser than lexical identities. For example, in the noun phrase (NP) chunking task, suppose that there are two lexical sequences, "*He is her –*" and "*He gave her –* ". The observed sequences, "*He is her*" and "*He gave her*", would both be conventionally labeled by '`BOB`', where `B` signifies the '*beginning NP*', and `O` the '*outside NP*'. However, this labeling may be too general to encapsulate their respective syntactic dynamics. In actuality, they have different latent-structures, crucial in labeling the next word. For "*He is her –*", the NP started by 'her' is still incomplete, so the label for – is likely to be `I`, which conveys the continuation of the phrase, e.g., "*[He] is [her brother]*". In contrast, for "*He gave her –*", the phrase started by '*her*' is normally self-complete, and makes the next label more likely to be `B`, e.g., "*[He] gave [her] [flowers]*".

In other words, latent-dynamics is an intermediate representation between input features and labels, and explicitly modeling this can simplify the problem. In particular, in many real-world cases, when the part-of-speech tags are not available, the modeling on latent-dynamics would be particularly important.

In this paper, we model latent-dynamics in shallow parsing by extending the Latent-Dynamic Conditional Random Fields (LDCRFs) (Morency et al. 2007), which offer advantages over previ-

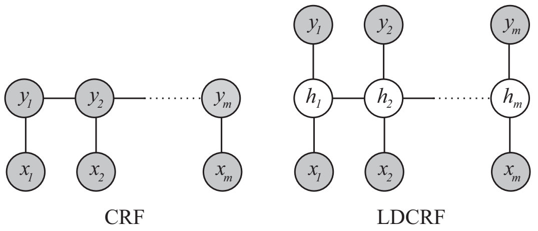

Figure 1: Comparison between CRF and LDCRF. In these graphical models, x represents the observation sequence, y represents labels and h represents hidden states assigned to labels. Note that only gray circles are observed variables. Also, only the links with the current observation are shown, but for both models, long range dependencies are possible.

ous learning methods by explicitly modeling hidden state variables (see Figure 1). We expect LDCRFs to be particularly useful in those cases without POS tags, though this paper is not limited to this.

The inference technique is one of the most important components for a structured classification model. In conventional models like CRFs, the optimal label path can be directly searched by using dynamic programming. However, for latent conditional models like LDCRFs, the inference is kind of tricky, because of hidden state variables. In this paper, we propose an exact inference algorithm, the Best Label Path inference, to efficiently produce the optimal label sequence on LDCRFs.

The following section describes the related work. We then review LDCRFs, and propose the BLP inference. We further present a statistical interpretation on learned hidden states. Finally, we show that LDCRF-BLP is particularly effective when pure word features are used, and when POS tags are added, as existing systems did, it achieves comparable results to the best reported systems.

2 Related Work

There is a wide range of related work on shallow parsing. Shallow parsing is frequently reduced to sequence labeling problems, and a large part of previous work uses machine learning approaches. Some approaches rely on k-order generative probabilistic models of paired input sequences and label sequences, such as HMMs (Freitag & McCallum 2000; Kupiec 1992) or multilevel Markov models (Bikel et al. 1999). The generative model provides well-understood training and inference but requires stringent conditional independence assumptions.

To accommodate multiple overlapping features on observations, some other approaches view the sequence labeling problem as a sequence of classification problems, including support vector machines (SVMs) (Kudo & Matsumoto 2001) and a variety of other classifiers (Punyakanok & Roth 2001; Abney et al. 1999; Ratnaparkhi 1996). Since these classifiers cannot trade off decisions at different positions against each other (Lafferty et al. 2001), the best classifier based shallow parsers are forced to resort to heuristic combinations of multiple classifiers.

A significant amount of recent work has shown the power of CRFs for sequence labeling tasks. CRFs use an exponential distribution to model the entire sequence, allowing for non-local dependencies between states and observations (Lafferty et al. 2001). Lafferty et al. (2001) showed that CRFs outperform classification models as well as HMMs on synthetic data and on POS tagging tasks. As for the task of shallow parsing, CRFs also outperform many other state-of-the-art models (Sha & Pereira 2003; McDonald et al. 2005).

When the data has distinct sub-structures, models that exploit hidden state variables are advantageous in learning (Matsuzaki et al. 2005; Petrov et al. 2007). Sutton et al. (2004) presented an extension to CRF called dynamic conditional random field (DCRF) model. As stated by the authors, training a DCRF model with unobserved nodes (hidden variables) makes their approach difficult to optimize. In the vision community, the LDCRF model was recently proposed by Morency et al. (2007), and shown to outperform CRFs, SVMs, and HMMs for visual sequence labeling.

In this paper, we introduce the concept of latent-dynamics for shallow parsing, showing how hidden states automatically learned by the model present similar characteristics. We will also propose an improved inference technique, the BLP, for producing the most probable label sequence in LDCRFs.

3 Latent-Dynamic Conditional Random Fields

The task is to learn a mapping between a sequence of observations $\mathbf{x} = x_1, x_2, \ldots, x_m$ and a sequence of labels $\mathbf{y} = y_1, y_2, \ldots, y_m$. Each y_j is a class label for the j'th token of a word sequence and is a member of a set $\mathbf{Y}$ of possible class labels. For each sequence, the model also assumes a vector of hidden state variables $\mathbf{y} = \{h_1, h_2, \ldots, h_m\}$, which are not observable in the training examples.

Given the above definitions, we define a latent conditional model as follows:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{h} P(\mathbf{y}|\mathbf{h}, \mathbf{x}, \mathbf{\Theta}) P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}), \qquad (1)$$

where $\mathbf{\Theta}$ are the parameters of the model. The LDCRF model can seem as a natural extension of the CRF model, and the CRF model can seem as a special case of LDCRFs employing one hidden state for each label.

To keep training and inference efficient, we restrict the model to have disjointed sets of hidden states associated with each class label. Each h_j is a member of a set $\mathbf{H}_{y_j}$ of possible hidden states for the class label y_j. We define $\mathbf{H}$, the set of all possible hidden states to be the union of all $\mathbf{H}_{y_j}$ sets. Since sequences which have any $\mathbf{h}_j \neq \mathbf{H}_{y_j}$ will by definition have $P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = 0$, we can express our model as:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h} \in \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}), \qquad (2)$$

where $P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta})$ is defined using the usual conditional random field formulation: $P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}) = \exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}|\mathbf{x}) / \sum_{\forall \mathbf{h}} \exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}|\mathbf{x})$, in which $\mathbf{f}(\mathbf{h}|\mathbf{x})$ is the feature vector. Given a training set consisting of n labeled sequences $(\mathbf{x}_i, \mathbf{y}_i)$ for $i = 1 \ldots n$, training is performed by optimizing the objective function to learn the parameter $\mathbf{\Theta}^*$:

$$L(\mathbf{\Theta}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{\Theta}) - R(\mathbf{\Theta}). \qquad (3)$$

The first term of this equation is the conditional log-likelihood of the training data. The second term is the regularizer.

4 BLP Inference on Latent Conditional Models

For testing, given a new test sequence x, we want to estimate the most probable label sequence (Best Label Path), $\mathbf{y}^*$, that maximizes our conditional model:

$$\mathbf{y}^* = \mathrm{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}^*). \qquad (4)$$

In the CRF model, $\mathbf{y}^*$ can be simply searched by using the Viterbi algorithm. However, for latent conditional models like LDCRF, the Best Label Path $\mathbf{y}^*$ cannot directly be produced by the Viterbi algorithm because of the incorporation of hidden states.

In this paper, we propose an exact inference algorithm, the Best Label Path inference (BLP), for producing the most probable label sequence $\mathbf{y}^*$ on LDCRF. In the BLP schema, top-n hidden paths $\mathbf{HP}_n = \{\mathbf{h}_1, \mathbf{h}_2 \ldots \mathbf{h}_n\}$ over hidden states are efficiently produced by using A^* search (Hart et al., 1968), and the corresponding probabilities of hidden paths $P(\mathbf{h}_i|\mathbf{x}, \mathbf{\Theta})$ are gained. Thereafter, based on $\mathbf{HP}_n$, the estimated probabilities of various label paths, $\overline{P}(\mathbf{y}|\mathbf{x}, \mathbf{\Theta})$, can be computed by summing the probabilities of hidden paths, $P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta})$, concerning the association between hidden states and each class label:

$$\overline{P}(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h}:\ \mathbf{h} \in \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m} \wedge \mathbf{h} \in \mathbf{HP}_n} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}). \qquad (5)$$

By using the A^* search, $\mathbf{HP}_n$ can be extended incrementally in an efficient manner, until the algorithm finds that the Best Label Path is ready, and then the search stops and ends the BLP inference with success. The algorithm judges that y^* is ready when the following condition is achieved:

$$\overline{P}(\mathbf{y}_1|\mathbf{x}, \mathbf{\Theta}) \leq \overline{P}(\mathbf{y}_2|\mathbf{x}, \mathbf{\Theta}) + \sum_{\mathbf{h} \notin \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}), \qquad (6)$$

where $\mathbf{y}_1$ is the most probable label sequence, and $\mathbf{y}_2$ is the second ranked label sequence estimated by using $\mathbf{HP}_n$. It would be straightforward to prove that $\mathbf{y}^* = \mathbf{y}_1$, and further search is unnecessary, because in this case, the unknown probability mass can not change the optimal label path. The unknown probability mass can be computed by using

$$\sum_{\mathbf{h} \notin \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}) = 1 - \sum_{\mathbf{h} \in \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}). \qquad (7)$$

The top-n hidden paths of $\mathbf{HP}_n$ produced by the A^*-search are exact, and the BLP inference is exact. To guarantee $\mathbf{HP}_n$ is exact in our BLP inference, an admissible heuristic function should be used in A^* search (Hart et al., 1968). We use a backward Viterbi algorithm (Viterbi, 1967) to compute the heuristic function of the forward A^* search:

$$\mathrm{Heu}_i(\mathbf{h}_j) = \max_{\mathbf{h}'_i = \mathbf{h}_j \wedge \mathbf{h}'_i \in \mathbf{HP}_i^{|\mathbf{h}|}} P'(\mathbf{h}'|\mathbf{x}, \mathbf{\Theta}^*), \qquad (8)$$

where $\mathbf{h}_i^{'} = \mathbf{h}_j$ represents a partial hidden path started from the hidden state $\mathbf{h}_j$, and $\mathbf{HP}_i^{|\mathbf{h}|}$ represents all possible partial hidden paths from the position i to the ending position $|\mathbf{h}|$. $\text{Heu}_i(\mathbf{h}_j)$ is an admissible heuristic function for the A^* search over hidden paths, therefore $\mathbf{HP}_n$ is exact and BLP inference is exact.

The BLP inference is efficient when the probability distribution among the hidden paths is intensive. By combining the forward A^* with the backward Viterbi algorithm, the time complexity of producing $\mathbf{HP}_n$ is roughly a linear complexity concerning its size. In practice, on the CoNLL test data containing 2,012 sentences, the BLP inference finished in five minutes when using the feature set based on both word and POS information (see Table 3). The memory consumption is also relatively small, because it is an online style algorithm and it is not necessary to preserve $\mathbf{HP}_n$.

In this paper, to make a comparison, we also study the Best Hidden Path inference (BHP):

$$\mathbf{y}_{BHP} = \text{argmax}_{\mathbf{y}} P(\mathbf{h}_{\mathbf{y}}|\mathbf{x}, \mathbf{\Theta}^*), \qquad (9)$$

where $\mathbf{h}_{\mathbf{y}} \in \mathbf{H}_{\mathbf{y}_1} \times \ldots \times \mathbf{H}_{\mathbf{y}_m}$. In other words, the Best Hidden Path is the label sequence that is directly projected from the most probable hidden path $\mathbf{h}^*$.

In (Morency et al. 2007), $\mathbf{y}^*$ is estimated by using the Best Point-wise Marginal Path (BMP). To estimate the label $\mathbf{y}_j$ of token j, the marginal probabilities $P(\mathbf{h}_j = a|\mathbf{x}, \mathbf{\Theta})$ are computed for possible hidden states $a \in \mathbf{H}$. Then, the marginal probabilities are summed and the optimal label is estimated by using the marginal probabilities.

The BLP produces y^* while the BHP and the BMP perform an estimation on y^*. We will make an experimental comparison in Section 6.

5 Analyzing Latent-Dynamics

The chunks in shallow parsing are represented with the three labels shown in Table 1, and shallow parsing is treated as a sequence labeling task with those three labels. A challenge for most shallow parsing approaches is to determine the concepts learned by the model. In this section, we show how we can analyze the latent-dynamics.

5.1 Analyzing Latent-Dynamics

In this section, we show how to analyze the characteristics of the hidden states. Our goal is to find the words characterizing a specific hidden state, and then look at the selected words with their associated POS tags to determine if the LDCRF model has learned meaningful latent-dynamics.

B	words beginning a chunk
I	words continuing a chunk
O	words being outside a chunk

Table 1: Shallow parsing labels.

In the experiments reported in this section, we did not use the features on POS tags in order to isolate the model's capability of learning latent dynamics. In other words, the model could simply learn the dynamics of POS tags as the latent dynamics if the model is given the information about POS tags. The features used in the experiments are listed on the left side (Word Features) in Table 3.

The main idea is to look at the marginal probabilities $P(\mathbf{h}_j = a|\mathbf{x}, \mathbf{\Theta})$ for each word j, and select the hidden state a^* with the highest probability. By counting how often a specific word selected a as the optimal hidden state, i.e., $\delta(w,\ a)$, we can create statistics about the relationship between hidden states and words. We define relative frequency as the number of times a specific word selected a hidden state while normalized by the global frequency of this word:

$$\text{RltFreq}(w, h_j) = \frac{\text{Freq}(\ \delta(w,\ h_j)\ )}{\text{Freq}(w)}. \qquad (10)$$

5.2 Learned Latent-Dynamics from CoNLL

In this subsection, we show the latent-dynamics learned automatically from the CoNLL dataset. The details of these experiments are presented in the following section.

The most frequent three words corresponding to the individual hidden states of the labels, B and O, are shown in Table 2. As shown, the automatically learned hidden states demonstrate prominent characteristics. The extrinsic label B, which begins a noun phrase, is automatically split into 4 subcategories: wh-determiners (*WDT*, such as "*that*") together with wh-pronouns (*WP*, such as "*who*"), the determiners (*DT*, such as "*any, an, a*"), the personal pronouns (*PRP*, such as "*they, we, he*"), and the singular proper nouns (*NNP*, such as "*Nasdaq, Florida*") together with the plural nouns (*NNS*, such as "*cities*"). The results of B1 suggests that the wh-determiners represented by "*that*", and the wh-pronouns represented by "*who*", perform simi-

Labels	HidStat	Words	POS	RltFreq
B	B1	*That*	*WDT*	0.85
		who	*WP*	0.49
		Who	*WP*	0.33
	B2	*any*	*DT*	1.00
		an	*DT*	1.00
		a	*DT*	0.98
	B3	*They*	*PRP*	1.00
		we	*PRP*	1.00
		he	*PRP*	1.00
	B4	*Nasdaq*	*NNP*	1.00
		Florida	*NNP*	0.99
		cities	*NNS*	0.99
O	O1	*But*	*CC*	0.88
		by	*IN*	0.73
		or	*IN*	0.67
	O2	4.6	*CD*	1.00
		1	*CD*	1.00
		11	*CD*	0.62
	O3	*were*	*VBD*	0.94
		rose	*VBD*	0.93
		have	*VBP*	0.92
	O4	*been*	*VBN*	0.97
		be	*VB*	0.94
		to	*TO*	0.92

Table 2: Latent-dynamics learned automatically by the LDCRF model. This table shows the top three words and their gold-standard POS tags for each hidden states.

lar roles in modeling the dynamics in shallow parsing. Further, the singular proper nouns and the plural nouns are grouped together, suggesting that they may perform similar roles. Moreover, we can notice that B2 and B3 are highly consistent.

The label O is automatically split into the coordinating conjunctions (*CC*) together with the prepositions (*IN*) indexed by O1, the cardinal numbers (*CD*) indexed by O2, the past tense verbs (*VBD*) together with the personal verbs (*VBP*) indexed by O3, and another sub-category, O4. From the results we can find that gold-standard POS tags may not be adequate in modeling latent-dynamics in shallow parsing, as we can notice that three hidden states out of four (O1, O3 and O4) contains relating but different gold-standard POS tags.

6 Experiments

Following previous studies on shallow parsing, our experiments are performed on the CoNLL 2000 data set (Sang & Buchholz 2000; Ramshow & Marcus 1995). The training set consists of 8,936 sentences, and the test set consists of 2,012 sentences. The standard evaluation metrics for this task are precision p (the fraction of output chunks matching the reference chunks), recall r (the fraction of reference chunks returned), and the F-measure given by $F = 2pr/(p + r)$.

Word Features:

$\{w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}, w_{i-1}w_i, w_iw_{i+1}\}$
$\times\{h_i, h_{i-1}h_i, h_{i-2}h_{i-1}h_i\}$

POS Features:

$\{t_{i-1}, t_i, t_{i+1}, t_{i-2}t_{i-1}, t_{i-1}t_i, t_it_{i+1}, t_{i+1}t_{i+2},$
$t_{i-2}t_{i-1}t_i, t_{i-1}t_it_{i+1}, t_it_{i+1}t_{i+2}\}$
$\times\{h_i, h_{i-1}h_i, h_{i-2}h_{i-1}h_i\}$

Table 3: Feature templates used in the experiments. w_i is the current word; t_i is current POS tag; and h_i is the current hidden state (for the case of latent models) or the current label (for the case of conventional models).

6.1 LDCRF for Shallow Parsing

We implemented LDCRFs in C++, and optimized the system to cope with large scale problems, in which the feature dimension is beyond millions. We employ similar predicate sets defined in Sha & Pereira (2003). We follow them in using predicates that depend on words as well as POS tags in the neighborhood of a given position, taking into account only those 417,835 features which occur at least once in the training data. The features are listed in Table 3.

As for numerical optimization (Malouf 2002; Wallach 2002), we performed gradient decent with the Limited-Memory BFGS (L-BFGS) optimization technique (Nocedal & Wright 1999). L-BFGS is a second-order Quasi-Newton method that numerically estimates the curvature from previous gradients and updates. With no requirement on specialized Hessian approximation, L-BFGS can handle large-scale problems in an efficient manner. We implemented an L-BFGS optimizer in C++ by modifying the OWLQN package (Andrew & Gao 2007) developed by Galen Andrew. In our experiments, storing 10 pairs of previous gradients for the approximation of the function's inverse Hessian worked well, making the amount of the extra memory required modest. Using more previous gradients will probably decrease the number of iterations required to reach convergence, but would increase memory requirements significantly. To make a comparison, we also employed the Conjugate-Gradient (CG) optimization algorithm. For details of CG, see Shewchuk (1994).

Since the objective function of the LDCRF model is non-convex, it is suggested to use the random initialization of parameters for the training. To reduce overfitting, we employed an L_2 Gaussian weight prior (Chen & Rosenfeld 1999). During training and validation, we varied the number of hidden states per label (from 2 to 6 states per label), and also varied the L_2-regularization term (with values 10^k, k from -3 to 3). Our experiments suggested that using 4 or 5 hidden states per label for the shallow parser is a viable compromise between accuracy and efficiency.

7 Results and Discussion

7.1 Performance on Word Features

As discussed in Section 4, it is preferred to not use the features on POS tags in order to isolate the model's capability of learning latent dynamics. In this sub-section, we use pure word features with their counts above 10 in the training data to perform experimental comparisons among different inference algorithms on LDCRFs, including BLP, BHP, and existing BMP.

Since the CRF model is one of the successful models in sequential labeling tasks (Lafferty et al. 2001; Sha & Pereira 2003; McDonald et al. 2005), in this section, we also compare LDCRFs with CRFs. We tried to make experimental results more comparable between LDCRF and CRF models, and have therefore employed the same features set, optimizer and fine-tuning strategy between LDCRF and CRF models.

The experimental results are shown in Table 4. In the table, Acc. signifies 'label accuracy', which is useful for the significance test in the following sub-section. As shown, LDCRF-BLP outperforms LDCRF-BHP and LDCRF-BMP, suggesting that BLP inference [1] is superior. The superiority of BLP is statistically significant, which will be shown in next sub-section. On the other side, all the LDCRF models outperform the CRF model. In particular, the gap between LDCRF-BLP and CRF is 1.53 percent.

Models: WF	Acc.	Pre.	Rec.	F_1
LDCRF-BLP	97.01	90.33	88.91	**89.61**
LDCRF-BHP	96.52	90.26	88.21	89.22
LDCRF-BMP	97.26	89.83	89.06	89.44
CRF	96.11	88.12	88.03	88.08

Table 4: Experimental comparisons among different inference algorithms on LDCRFs, and the performance of CRFs using the same feature set on pure word features. The BLP inference outperforms the BHP and BMP inference. LDCRFs outperform CRFs.

Models	F_1 Gap	Acc. Gap	Sig.
BLP vs. BHP	0.39	0.49	1e-10
BLP vs. CRF	1.53	0.90	5e-13

Table 5: The significance tests. LDCRF-BLP is significantly more accurate than LDCRF-BHP and CRFs.

7.2 Labeling Accuracy and Significance Test

As shown in Table 4, the accuracy rate for individual labeling decisions is over-optimistic as a measure for shallow parsing. Nevertheless, since testing the significance of shallow parsers' F-measures is tricky, individual labeling accuracy provides a more convenient basis for statistical significance tests (Sha & Pereira 2003). One such test is the McNemar test on paired observations (Gillick & Cox 1989). As shown in Table 5, for the LDCRF model, the BLP inference schema is statistically more accurate than the BHP inference schema. Also, Evaluations show that the McNemar's value on labeling disagreement between the LDCRF-BLP and CRF models is 5e-13, suggesting that LDCRF-BLP is significantly more accurate than CRFs.

On the other hand, the accuracy rate of BMP inference is a special case. Since the BMP inference is essentially an accuracy-first inference schema, the accuracy rate and the F-measure have a different relation in BMP. As we can see, the individual labeling accuracy achieved by the LDCRF-BMP model is as high as 97.26%, but its F-measure is still lower than LDCRF-BLP.

7.3 Convergence Speed

It would be interesting to compare the convergence speed between the objective loss function of LDCRFs and CRFs. We apply the L-BFGS optimiza-

[1] In practice, for efficiency, we approximated the BLP on a few sentences by limiting the number of search steps.

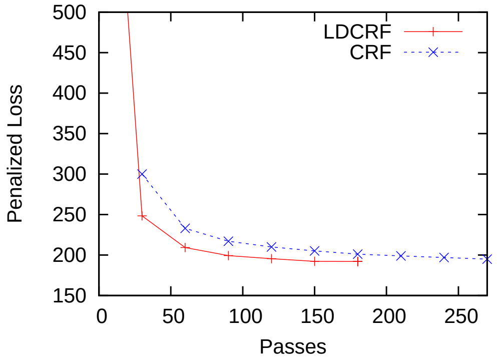

Figure 2: The value of the penalized loss based on the number of iterations: LDCRFs vs. CRFs.

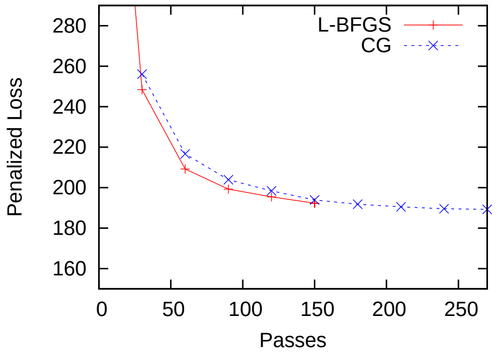

Figure 3: Training the LDCRF model: L-BFGS vs. CG.

tion algorithm to optimize the loss function of LDCRF and CRF models, making a comparison between them. We find that the iterations required for the convergence of LDCRFs is less than for CRFs (see Figure 2). Normally, the LDCRF model arrives at the plateau of convergence in 120-150 iterations, while CRFs require 210-240 iterations. When we replace the L-BFGS optimizer by the CG optimization algorithm, we observed as well that LDCRF converges faster on iteration numbers than CRF does.

On the contrary, however, the time cost of the LDCRF model in each iteration is higher than the CRF model, because of the incorporation of hidden states. The time cost of the LDCRF model in each iteration is roughly a quadratic increase concerning the increase of the number of hidden states. Therefore, though the LDCRF model requires less passes for the convergence, it is practically slower than the CRF model. Improving the scalability of the LDCRF model would be a interesting topic in the future.

Furthermore, we make a comparison between the L-BFGS and the CG optimizer for LDCRFs. We observe that the L-BFGS optimizer is slightly faster than CG on LDCRFs (see Figure 3), which echoes the comparison between the L-BFGS and the CG optimizing technique on the CRF model (Sha & Pereira 2003).

Models: WF+POS	Pre.	Rec.	F_1
LDCRF-BLP	94.65	94.03	**94.34**
CRF (Vishwanathan et al. 06)	N/A	N/A	93.6
CRF (McDonald et al. 05)	94.57	94.00	94.29
Voted perceptron (Collins 02)	N/A	N/A	93.53
Generalized Winnow (Zhang et al. 02)	93.80	93.99	93.89
SVM combination (Kudo & Matsumoto 01)	94.15	94.29	94.22
Memo. classifier (Sang 00)	93.63	92.89	93.26

Table 6: Performance of the LDCRF-BLP model, and the comparison with CRFs and other successful approaches. In this table, all the systems have employed POS features.

7.4 Comparisons to Other Systems with POS Features

Performance of the LDCRF-BLP model and some of the best results reported previously are summarized in Table 6. Our LDCRF model achieved comparable performance to those best reported systems in terms of the F-measure.

McDonald et al. (2005) achieved an F-measure of 94.29% by using a CRF model. By employing a multi-model combination approach, Kudo & Matsumoto (2001) also achieved a good performance. They use a combination of 8 kernel SVMs with a heuristic voting strategy. An advantage of LDCRFs over max-margin based approaches is that LDCRFs can output N-best label sequences and their probabilities using efficient marginalization operations, which can be used for other components in an information extraction system.

8 Conclusions and Future Work

In this paper, we have shown that automatic modeling on "latent-dynamics" can be useful in shallow parsing. By analyzing the automatically learned hidden states, we showed how LDCRFs can naturally learn latent-dynamics in shallow parsing.

We proposed an improved inference algorithm, the BLP, for LDCRFs. We performed experiments using the CoNLL data, and showed how the BLP inference outperforms existing inference engines. When further employing POS features as other systems did, the performance of the LDCRF-BLP model is comparable to those best reported results. The LDCRF model demonstrates a significant advantage over other models on pure word features in this paper. We expect it to be particularly useful in the real-world tasks without rich features.

The latent conditional model handles latent-dynamics naturally, and can be easily extended to other labeling tasks. Also, the BLP inference algorithm can be extended to other latent conditional models for producing optimal label sequences. As a future work, we plan to further speed up the BLP algorithm.

Acknowledgments

Many thanks to Yoshimasa Tsuruoka for helpful discussions on the experiments and paper-writing. This research was partially supported by Grant-in-Aid for Specially Promoted Research 18002007 (MEXT, Japan). The work at the USC Institute for Creative Technology was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and the content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Abney, S. 1991. Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, Principle-based Parsing. *Kluwer Academic Publishers.*

Abney, S.; Schapire, R. E. and Singer, Y. 1999. Boosting applied to tagging and PP attachment. In Proc. *EMNLP/VLC-99.*

Andrew, G. and Gao, J. 2007. Scalable training of L1-regularized log-linear models. In Proc. *ICML-07.*

Bikel, D. M.; Schwartz, R. L. and Weischedel, R. M. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34: 211-231.

Chen, S. F. and Rosenfeld, R. 1999. A Gaussian prior for smooth-ing maximum entropy models. *Technical Report CMU-CS-99-108*, CMU.

Collins, M. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In Proc. *EMNLP-02.*

Freitag, D. and McCallum, A. 2000. Information extraction with HMM structures learned by stochastic optimization. In Proc. *AAAI-00.*

Gillick, L. and Cox, S. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics Speech and Signal Processing*, v1, pages 532-535.

Hart, P.E.; Nilsson, N.J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost path. IEEE Trans. On *System Science and Cybernetics*, SSC-4(2): 100-107.

Kudo, T. and Matsumoto, Y. 2001. Chunking with support vector machines. In Proc. *NAACL-01.*

Kupiec, J. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language.* 6:225-242.

Lafferty, J.; McCallum, A. and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. *ICML-01*, pages 282-289.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In Proc. *CoNLL-02.*

Matsuzaki, T.; Miyao Y. and Tsujii, J. 2005. Probabilistic CFG with Latent Annotations. In Proc. *ACL-05.*

McDonald, R.; Crammer, K. and Pereira, F. 2005. Flexible Text Segmentation with Structured Multilabel Classification. In Proc. *HLT/EMNLP-05*, pages 987- 994.

Morency, L.P.; Quattoni, A. and Darrell, T. 2007. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In Proc. *CVPR-07*, pages 1- 8.

Nocedal, J. and Wright, S. J. 1999. Numerical Optimization. *Springer.*

Petrov, S.; Pauls, A.; and Klein, D. 2007. Discriminative log-linear grammars with latent variables. In Proc. *NIPS-07.*

Punyakanok, V. and Roth, D. 2001. The use of classifiers in sequential inference. In Proc. *NIPS-01*, pages 995-1001. MIT Press.

Ramshaw, L. A. and Marcus, M. P. 1995. Text chunking using transformation-based learning. In Proc. Third Workshop on Very Large Corpora. In Proc. *ACL-95.*

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In Proc. *EMNLP-96.*

Sang, E.F.T.K. 2000. Noun Phrase Representation by System Combination. In Proc. *ANLP/NAACL-00.*

Sang, E.F.T.K and Buchholz, S. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In Proc. *CoNLL-00*, pages 127-132.

Sha, F. and Pereira, F. 2003. Shallow Parsing with Conditional Random Fields. In Proc. *HLT/NAACL-03.*

Shewchuk, J. R. 1994. An introduction to the conjugate gradient method without the agonizing pain. http://www.2.cs.cmu.edu/jrs/jrspapers.html/#cg.

Sutton, C.; Rohanimanesh, K. and McCallum, A. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In Proc. *ICML-04.*

Viterbi, A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory.* 13(2):260-269.

Vishwanathan, S.; Schraudolph, N. N.; Schmidt, M.W. and Murphy, K. 2006. Accelerated training of conditional random fields with stochastic meta-descent. In Proc. *ICML-06.*

Wallach, H. 2002. Efficient training of conditional random fields. In *Proc. 6th Annual CLUK Research Colloquium.*

Zhang, T.; Damerau, F. and Johnson, D. 2002. Text chunking based on a generalization of winnow. *Journal of Machine Learning Research*, 2:615-637.