Sequential Labeling Using Deep-Structured Conditional Random Fields

Dong Yu, *Senior Member, IEEE*, Shizhen Wang, and Li Deng, *Fellow, IEEE*

***Abstract*—We develop and present the deep-structured conditional random field (CRF), a multi-layer CRF model in which each higher layer's input observation sequence consists of the previous layer's observation sequence and the resulted frame-level marginal probabilities. Such a structure can closely approximate the long-range state dependency using only linear-chain or zeroth-order CRFs by constructing features on the previous layer's output (belief). Although the final layer is trained to maximize the log-likelihood of the state (label) sequence, each lower layer is optimized by maximizing the frame-level marginal probabilities. In this deep-structured CRF, both parameter estimation and state sequence inference are carried out efficiently layer-by-layer from bottom to top. We evaluate the deep-structured CRF on two natural language processing tasks: search query tagging and advertisement field segmentation. The experimental results demonstrate that the deep-structured CRF achieves word labeling accuracies that are significantly higher than the best results reported on these tasks using the same labeled training set.**

***Index Terms*—Conditional random fields (CRFs), deep-structure, marginal probability, natural language processing, sequential labeling, word tagging.**

I. Introduction

CONDITIONAL random fields (CRFs) have been successfully applied to sequential labeling problems, notably those in natural language processing applications, for several years [9], [15], [16], [20], [29]. Unlike the hidden Markov model (HMM), a *generative* model that describes the joint probability of the observation data and the class labels, CRFs are *discriminative* models that estimate the class label sequence conditional probabilities directly. In the HMMs, observations in different frames (e.g., word tokens at different positions) are assumed to be independent given the state. However, CRFs do not require this assumption and hence have high flexibility in choosing features, including those that may not exist in some frames (i.e., word positions in the natural language processing tasks) and those that depend on the entire observation sequence.

The most popular CRF for sequential labeling is the linear-chain CRF depicted in Fig. 1 due to its simplicity and efficiency. Let us denote by $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ the T-frame observation sequence, and by $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ the corresponding state (label) sequence, which can be augmented with a special start (y_0) and end (y_{T+1}) state.

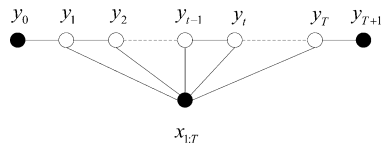


Fig. 1. Graphical representation of the linear-chain CRF, where $\boldsymbol{x}$ is the observation sequence and $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ is the label sequence. The solid and empty nodes denote the observed and unobserved variables, respectively.

In the linear-chain CRFs, the conditional probability of a state (label) sequence $\boldsymbol{y}$ given the observation sequence $\boldsymbol{x}$ is given by

$$p(\boldsymbol{y} \,|\, \boldsymbol{x}; \Lambda) = \frac{\exp\left(\sum_{t,i} \lambda_i f_i(y_t, y_{t-1}, \boldsymbol{x}, t)\right)}{Z(\boldsymbol{x}; \Lambda)} \tag{1}$$

where we have used $f_i(y_t, y_{t-1}, \boldsymbol{x}, t)$ to represent both the observation features $f_i(y_t, \boldsymbol{x}, t)$ that provide constraints between the observation sequence and the state at time t, and the state transition features $f(y_t, y_{t-1}, t)$ that provide constraints on the consecutive states. $\Lambda = \{\lambda_i\}$ are the model parameters, and

$$Z(\boldsymbol{x}; \Lambda) = \sum_{\boldsymbol{y}} \exp\left(\sum_{t,i} \lambda_i f_i(y_t, y_{t-1}, \boldsymbol{x}, t)\right) \tag{2}$$

is the partition function to normalize the exponential form so that it becomes a valid probability measure.

The model parameters Λ in the linear-chain CRFs are typically optimized to maximize the L_2 regularized state sequence log-likelihood

$$J_1(\Lambda, X) = \sum_k \log p\left(\boldsymbol{y}^{(\boldsymbol{k})} \,\middle|\, \boldsymbol{x}^{(\boldsymbol{k})}; \Lambda\right) - \frac{\|\Lambda\|^2}{2\sigma^2} \tag{3}$$

where σ^2 is a parameter that balances the log-likelihood and the regularization term and can be tuned using a development set. The derivatives of $J_1(\Lambda, X)$ over the model parameters λ_i are given by

$$\begin{aligned}\frac{\partial J_1(\Lambda, X)}{\partial \lambda_i} &= \tilde{E}[f_i(\boldsymbol{y}, \boldsymbol{x})] - E[f_i(\boldsymbol{y}, \boldsymbol{x})] - \frac{\lambda_i}{\sigma^2} \\ &= \sum_k f_i\left(\boldsymbol{y}^{(\boldsymbol{k})}, \boldsymbol{x}^{(\boldsymbol{k})}\right) - \frac{\lambda_i}{\sigma^2} \\ &\quad - \sum_k \sum_{\boldsymbol{y}} p\left(\boldsymbol{y} \,\middle|\, \boldsymbol{x}^{(\boldsymbol{k})}; \Lambda\right) f_i\left(\boldsymbol{y}, \boldsymbol{x}^{(\boldsymbol{k})}\right). \end{aligned} \tag{4}$$

The parameters in the linear-chain CRFs can be efficiently estimated using the forward–backward (sum–product) algorithm [3] along with the optimization algorithms such as generalized iterative scaling (GIS) [6], gradient ascent, quasi-Newton

Manuscript received August 17, 2009; accepted February 19, 2010. Date of publication September 13, 2010; date of current version November 17, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong He.

D. Yu and L. Deng are with Microsoft Research, Redmond, WA 98034 USA (e-mail: dongyu@microsoft.com; deng@microsoft.com).

S. Wang was with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA. He is now with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: shizhen@microsoft.com).

Digital Object Identifier 10.1109/JSTSP.2010.2075990

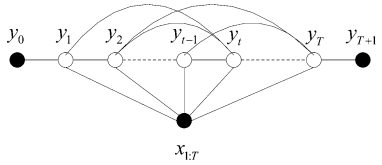


Fig. 2. Graphical representation of a complicated CRF in which a state may have links to several distant states. The solid and empty nodes denote the observed and unobserved variables, respectively.

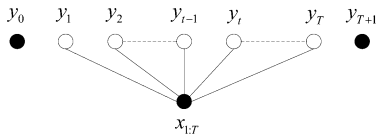


Fig. 3. Graphical representation of a CRF without state transition features. Such a zeroth-order CRF is extremely efficient in parameter estimation and state sequence inference. The solid and empty nodes denote the observed and unobserved variables, respectively. For consistency, we keep the start and end states in the graph, although removing them makes no difference to the model.

method (e.g., L-BFGS [19]), conjugate gradient approach, and resilient propagation (RPROP) [21]. The state sequence inference problem can be efficiently solved using the Viterbi (max-product) algorithm [3].

In the linear-chain CRF, the constraints between skip-states are indirectly modeled by the constraints between the consecutive states. More complicated CRFs can be used to impose direct and stronger constraints between the skip-states. For example, Fig. 2 depicts such a CRF, in which a state may have links to several distant states. Although improved performance can be obtained with these more complicated CRFs over the linear-chain CRF, the cost associated with the parameter estimation and state sequence inference problems in these models is substantially higher. In many cases, approximation methods such as loopy belief propagation and variational methods [3], [22] are needed to make both parameter learning and state sequence inference tractable.

We take a different approach in this work and develop and present a deep-structured CRF. Instead of using more complicated CRFs, we use multiple layers of simple CRFs, including the linear-chain CRF (Fig. 1) and even the zeroth-order CRF (Fig. 3) that does not use state transition features. We demonstrate that we can greatly increase the modeling power using our proposed framework without substantially sacrificing the efficiency.

Different from other hierarchical, CRFs in the literature [17], [13], [23], which aim at tackling the granularity problem at different representation layers and use the lower layer CRFs as the building blocks for the higher layer CRFs, in our model the observation sequence of each higher layer CRF consists of the previous layer's observation sequence and the resulted frame-level marginal posterior probabilities, as will be described in detail in Section II. Since the features in the CRF can be constructed on the entire observation sequence, the features in the higher layer CRFs can be constructed upon the lower-layer beliefs (posterior probabilities) of the frames that are farther away from the current frame. In other words, it can approximate the long-range state dependency using beliefs from lower layers, which was shown in our earlier work to be helpful in improving the inference accuracy in speech recognition [7], [25], [26], [7] and natural language processing whose time series data have rich structures with cues spanning over long ranges of the decoded states. The prior work that is closest to our approach is the stacked sequential learning proposed by Cohen and Carvalho [5]. Our work extends their work in the adoption of different training criteria at different layers and the ability to learn the intermediate hidden layers in an unsupervised way [30], [31].

Both parameter estimation and state sequence inference in the deep-structured CRF are carried out in a layer-by-layer fashion from bottom to top. More specifically, during the parameter estimation stage, each layer is trained independently instead of jointly with the lower layers trained first. This strategy guarantees the efficiency of parameter estimation and state sequence inference, with a time complexity linear to the number of layers used. Although not the focus of this paper, it is important to note that the intermediate layers can be considered as internal representations of the original observation at different granularities and so each intermediate layer may have different numbers of states. For example, when applied to the language identification task, the number of states in the first and second layer CRFs can be 128 and 7, respectively, as described in the related work [31]. In this paper, however, we have assumed that the numbers of states in the intermediate layers are the same as that in the final layer and so the supervision for the intermediate layers can be taken from the final layer during the training process. Learning with hidden states in the deep-structured CRF is more challenging. Interested readers can find an effective solution in our recent work [30], [31].

We have evaluated the deep-structured CRF on two sequential labeling tasks in natural language processing: the search query tagging task and the advertisement field segmentation task. We investigate the strengths and weaknesses of the deep-structured CRF under various structures. The results presented in Section III demonstrate that the deep-structured CRF achieves word labeling accuracies significantly higher than the best results reported on the same task using the identical labeled training set.

The rest of the paper is organized as follows. In Section II, we describe the architecture of the deep-structured CRF model, study the properties of the model, and introduce the frame-level maximum marginal log-likelihood criterion for optimizing the intermediate layers of the model. In Section III, we present experimental results on two natural language processing tasks using the deep-structured CRF model, analyze the empirical performance of the model, and demonstrate the effectiveness of the model via comparisons with other approaches on the identical tasks and with identical training data. We conclude the paper in Section IV.

II. Deep-Structured CRF

In this section, we describe the architecture and properties of the deep-structured CRF model in detail. In particular, we establish distinct training criteria used to learn the parameters at different layers of the deep-structured CRF model.

A. Architecture of the Deep-Structured CRF

The architecture of the deep-structured CRF model developed and evaluated in this work is shown in Fig. 4. It consists of a hi-

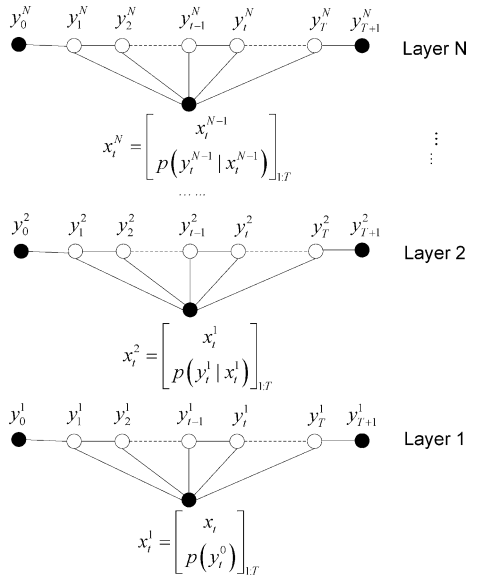


Fig. 4. Graphical representation of the deep-structured CRF in which each higher layer's input observation sequence consists of the observation sequence and the frame-level marginal posterior probabilities from the previous layer. The solid and empty nodes denote the observed and unobserved variables, respectively, and the connections between states are optional.

erarchy of linear-chain CRFs and/or zeroth-order CRFs that do not use state transition features. The augmented observation sequence of layer j contains both the previous layer's observation sequence $\boldsymbol{x^{j-1}}$ and the frame-level marginal posterior probabilities $p(y_t^{j-1} \mid \boldsymbol{x^{j-1}})$ from the preceding layer $j-1$. Note in Fig. 4 $p(y_t^0)$ is the prior probability of the labels and can be set to the uniform distribution, or removed from the first layer's input if the prior information is not available. We want to point out that the approach of constructing the observation sequence described in this section above is similar to the tandem structure [10] used in the automatic speech recognition systems where the frame-level sub-phone posterior probabilities generated by a neural network, together with the original observation feature vector, are fed into the HMM as the input. This approach guarantees that no information will be lost after each processing step. Since arbitrary features can be constructed on the entire observation sequence the higher layer CRFs can make use of the posterior probabilities (or beliefs) of frames farther away from the current frame and approximate the long-range dependencies. It is noted that although marginal posteriors from all previous layers are available to construct features, they are not necessarily all used. In fact, only the original observation sequence and the immediate previous layer's output are needed for most applications. In addition, feature selection techniques can be used to automatically determine the features (e.g., the observation sequences used and the range of the dependencies needed) for any particular task.

Both model parameter estimation and state sequence inference are carried out layer-by-layer in a bottom-up manner, and throughout this paper we use the supervised learning paradigm where both the observation sequence and the corresponding labels are available. During the parameter estimation or training/learning stage, once a lower layer CRF is trained, the model parameters of that layer are fixed and the corresponding frame-level marginal posterior probabilities are obtained and used as part of the observations fed to the next layer. This process continues until the model parameters of the highest or final layer of the model are optimized. The inference process is similar. The original observation is first processed by the bottom layer and the generated frame-level marginal posterior probabilities are fed into the next layer together with the previous layer's observation sequence. This process continues up to the highest layer of the model, the output of which is the inferred state sequence instead of the marginal probabilities. By learning and making inference on each layer independently, instead of jointly, we can limit the computational complexity to be at most linear to the number of layers used.

While the number of states or labels at the highest layer is determined by the problem to be solved, the number of states at other layers can be arbitrary and does not need to be the same across layers. In fact, each intermediate layer can be considered as an abstract internal representation of the original observation at different granularities and can be estimated using either unsupervised or supervised approaches. For example, a simple approach to determining the number of states in the intermediate layers is to cluster the observations using Gaussian mixture models. Another approach is to learn the intermediate layers so that the original observation can be reconstructed from the internal representations. In [30], we described an approach with which the intermediate layers are first pre-trained to maximize the state occupation entropy and minimize the frame-level conditional entropy at the same time and then fine-tuned using the back-propagation algorithm.

In this paper, to limit the scope, we will not discuss the learning of the hidden intermediate representations. Significant performance improvements have already been achieved on the sequential labeling tasks (which we will discuss in Section III) by assuming the intermediate representation to be the same as the final representation. By using the same state values in all layers, training of each layer can be carried out in efficient supervised way since we can use the same label sequence as the supervision in each layer.

B. Optimization Criteria and Techniques

The output of the deep-structured CRF model is state sequences. For this reason, the parameters in the highest or final layer of the model is optimized by maximizing the regularized log-likelihood (3) at the state-sequence level. In contrast to the highest layer, all remaining layers are trained by maximizing the frame-level marginal log-likelihood of

$$J_2(\Lambda, X) = \sum_{k,t} \log p\left(y_t^{(k)} \,\middle|\, \boldsymbol{x}^{(k)}; \Lambda\right) - \frac{||\Lambda||^2}{2\sigma^2} \qquad (5)$$

since it is this conditional marginal probability that is passed into the higher layers. Optimizing the frame-level conditional marginal probability at lower layers allows for those layers to make more accurate estimation of the state the frame should be in (rather than the sequence should be in) and thus can provide better information to the higher layers which make decisions on longer ranges. In Section III, we will show empirically that

this optimization criterion can make a difference in the state sequence inference accuracy.

The key to optimizing $J_2(\Lambda, X)$ is to compute the derivative of

$$\begin{aligned}\frac{\partial J_2(\Lambda,X)}{\partial \lambda_i} &= \sum_{k,t} \frac{\partial \log p\left(y_t^{(k)} \,\middle|\, \boldsymbol{x}^{(k)};\Lambda\right)}{\partial \lambda_i} - \frac{\lambda_i}{\sigma^2} \\ &= \sum_{k,t} \frac{\partial \log p\left(y_t^{(k)}, \boldsymbol{x}^{(k)};\Lambda\right)}{\partial \lambda_i} \\ &\quad - \sum_{k,t} \frac{\partial \log Z\left(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda\right)}{\partial \lambda_i} - \frac{\lambda_i}{\sigma^2}. \end{aligned} \tag{6}$$

Note that $\partial Z(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda)/\partial\lambda_i$ can be efficiently computed with the same approach as used in optimizing $J_1(\Lambda, X)$, and

$$\frac{\partial \log p\left(y_t^{(k)}, \boldsymbol{x}^{(k)};\Lambda\right)}{\partial \lambda_i} = \frac{\partial p\left(y_t^{(k)}, \boldsymbol{x}^{(k)};\Lambda\right)/\partial\lambda_i}{p\left(y_t^{(k)}, \boldsymbol{x}^{(k)};\Lambda\right)} \tag{7}$$

where

$$\frac{\partial p\left(y_t^{(k)}, \boldsymbol{x}^{(k)};\Lambda\right)}{\partial \lambda_i} = \sum_{\boldsymbol{y}\backslash y_t = y_t^{(k)}} \frac{\partial p\left(\boldsymbol{y}, \boldsymbol{x}^{(k)};\Lambda\right)}{\partial \lambda_i} \tag{8}$$

and $\boldsymbol{y} \setminus y_t = y_t^{(k)}$ denotes the summation over all possible state sequences with the state of the tth frame clamped to $y_t^{(k)}$. Equation (8) can be efficiently evaluated with a forward–backward algorithm that is similar to the one used in computing $\partial Z(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda)/\partial\lambda_i$.

In the zeroth-order CRF where transition features are not used (Fig. 3), optimizing $J_2(\Lambda, X)$ is equivalent to optimizing $J_1(\Lambda, X)$ since

$$\begin{aligned} J_1(\Lambda,X) &= \sum_k \log p\left(\boldsymbol{y}^{(\boldsymbol{k})} \,\middle|\, \boldsymbol{x}^{(\boldsymbol{k})};\Lambda\right) - \frac{\|\Lambda\|^2}{2\sigma^2} \\ &= \sum_k \log \frac{\exp\left(\sum_{t,i} \lambda_i f_i\left(y_t^{(\boldsymbol{k})}, y_{t-1}^{(\boldsymbol{k})}, \boldsymbol{x}^{(\boldsymbol{k})}, t\right)\right)}{Z\left(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda\right)} \\ &\quad - \frac{\|\Lambda\|^2}{2\sigma^2} \\ &= \sum_k \sum_{t,i} \lambda_i f_i\left(y_t^{(\boldsymbol{k})}, \boldsymbol{x}^{(\boldsymbol{k})}, t\right) - \log Z\left(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda\right) \\ &\quad - \frac{\|\Lambda\|^2}{2\sigma^2} \\ &= \sum_{k,t} \sum_{i} \lambda_i f_i\left(y_t^{(\boldsymbol{k})}, \boldsymbol{x}^{(\boldsymbol{k})}, t\right) - \log Z\left(\boldsymbol{x}^{(\boldsymbol{k})};\Lambda\right) \\ &\quad - \frac{\|\Lambda\|^2}{2\sigma^2} \\ &= J_2(\Lambda, X). \end{aligned} \tag{9}$$

In fact, training and inference for the zeroth-order CRF have the complexity of $O(TY)$, where T is the number of frames and Y is the number of states. This is significantly better than the time complexity of $O(TY^2)$ when the transition features are used. Even more attractive about the zeroth-order CRF is that the output of each frame is independent of each other and so the process can be further speeded up using parallel computing techniques.

C. Properties

We now discuss some properties associated with the deep-structured CRF.

Theorem 1: The objective function $J_1(\Lambda, X)$ on the training set will not decrease as more layers are added in the deep-structured CRF.

Proof: Let us consider the extension from an N-layer deep-structured CRF to an $N+1$ layer deep-structured CRF. The parameters for the first $N-1$ layers are the same for both systems. For the N-layer system, the observation features at the final layer are constructed on $\boldsymbol{x}^{\boldsymbol{N}}$ and the corresponding parameter set is Λ^N. For the $N+1$-layer system, the observation features are constructed on the observations that are augmented by $p'(y_t^N \,|\, \boldsymbol{x}^{\boldsymbol{N}})$ at each frame, where we use p' to indicate that the probability is estimated using the Nth layer in the $N+1$-layer system. The corresponding parameter set at the final layer in the $N+1$-layer system is $\Lambda^{N+1} \supseteq \Lambda^N$. Since

$$\max_{\Lambda^N} J_1(\Lambda^N, X) \le \max_{\Lambda^{N+1}} J_1(\Lambda^{N+1}, X) \tag{10}$$

and the optimization problem is convex at each layer, the learning algorithm can always find a parameter set in the $N+1$-layer system that gives a higher value of $J_1(\Lambda, X)$. ■

It directly follows that

Corollary 1: The deep-structured CRF performs no worse than the single-layer linear-chain CRF on the training set.

Note that the conditional log-likelihood increase in the training set can be carried over to the test set with a properly chosen regularization term. However, as the number of layers continues to grow, the gain will eventually saturate on the test set.

Look from a different angle, if we restrict that only the original observation sequence and the immediate previous layer's output are used as observations in higher layers, we can use the same feature set for all the layers other than the first layer in which additional observation features $f_k^j(y_t, \boldsymbol{x}, t) = p(y_{t+k}^{j-1} \,|\, \boldsymbol{x}^{\boldsymbol{j-1}})$ (k is the relative frame number in the previous layer) are not used. If we further restrict that each layer uses the same weights, the deep-structured CRF performs the same as iterative approaches (such as the one described in [22]), where the beliefs are updated and fed back as features. The number of iterations in the iterative approach is equivalent to the number of layers in the deep-structured CRF. If we allow for infinite number of layers (or iterations), the deep-structured CRF essentially simulates the (loopy) belief propagation in which the additional observation features describe the higher order state constraints and the beliefs at each frame are scheduled to be updated simultaneously based on previous layer's beliefs.

Since we are not restricted to use the same weights and same features at different layers (e.g., we can use features constructed from more frames at higher layers) the deep-structured CRF is more flexible, easier to train and has potential to achieve good performance with less layers (or iterations).

Note that when continuous features (e.g., mel-frequency cepstrum coefficient in speech processing systems) are used and sufficient training data are available, better performance can be achieved by expanding each continuous feature into several continuous features as discussed in [28] with the cubic spline approximation techniques developed in [25] and [29]. The core idea of the technique is to use distribution constraints instead of mean constraints in the system and thus the information contained in the feature distribution can be utilized to improve the sequential labeling accuracy.

III. Experimental Evaluation

To better understand the properties of the deep-structured CRF model and to study the strengths and weaknesses of different structures, we have conducted a series of experiments on two natural language processing tasks: a search query tagging task and an advertisement field segmentation task. Through the empirical evaluation using these tasks, we also show the performance gain obtained by using the deep-structured CRF compared with the conventional single-layered linear-chain CRF, and demonstrate the ability of the deep-structured CRF to achieve the best word labeling accuracy without using complicated features or additional external knowledge.

Both tasks discussed in this section share the same theme: extract information from the web-related texts written in natural languages. This is an area that gains great interests recently [1], [2], [4], [9], [14]–[16], [18], [20], [24], [29], [33]. The successful application of our model to these tasks thus has practical significance. Since the lengths of search queries are very short (as described in Section III-A) some properties of our model may not manifest on this task. For this reason, we examine more carefully the performance on the advertisement field segmentation task.

In all the experiments we have conducted, the regularization terms are determined and the models are selected based on performance of the development sets. The RPROP [21] training algorithm was used in all the experiments to learn the model parameters with the initial parameters set to zero and the initial RPROP step size set to the gradient under the initial parameter configuration. The batch sizes used in the experiments were initially set to one and were increased by 1.2 times iteration by iteration. The training algorithm stops when either the maximum number of iterations is reached or the log-likelihood gain over the iteration is less than a preset threshold.

A. Task of Product Search Query Tagging

In the search query tagging task, each query is a sequence of word tokens. Our goal is to assign a label from a set of predefined fields (or tags) to each word token. More specifically, we focus on tagging product search queries with the nine fields defined in Table I. The evaluation metric used is the word labeling accuracy (WLA) which is the percentage of word tokens that are correctly tagged.

We have used the computing electronics search query data collected from a three-month query log of www.live.com. This is the same data set as used by the earlier published work of Li *et al.* [15], except that we used only the manually labeled queries as the training data and Li *et al.* also used 250 K queries with derived labels for training. Additional information on how the data were collected, processed, and labeled can be found in [15]. The data set contains 15 K manually labeled training queries and 700 test queries (or 2 K word tokens) with three words on average in each query. From the 15 K training queries, we separated 700 of them as the development set. It has been reported in [15] that for this data set the inter-rater agreement of the labels between different annotators is around 80% at the query level and 91% at the token level. This marks the best meaningful accuracy we can obtain on this task.

TABLE I
Fields Used in the Product Search Query Tagging Task

Fields	Example
Brand	**Honda** civic
Model	Honda **civic**
Type	**Mp3 player**
Attribute	Honda civic **silver**
Merchant	Mp3 player at **best buy**
SortOrder	**best** mp3 players
BuyingIntent	**buy** mp3 players
ResearchIntent	mp3 player **review**
Other	Mp3 player **at** best buy

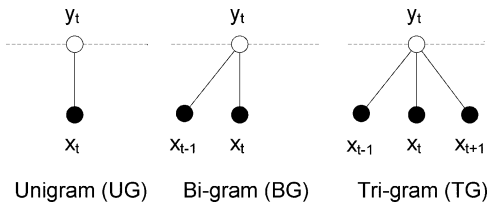


Fig. 5. Illustration of unigram, bi-gram, and tri-gram features constructed from the observation vectors.

We used a straightforward feature extraction technique in our experiments. We first built a lexicon that contains all words observed in the training set. Using this lexicon, each word in both training and test sets can be mapped into a token ID. The observation vector at each frame has the dimensionality equal to the lexicon size. The vector component takes a value of one corresponding to the token ID observed at the frame. It takes a value of zero at all other vector components. If the word is not in the lexicon, the observation vector takes a value of zero at all vector components, i.e.,

$$x_{t,i} = \begin{cases} 1, & \text{if } i = \text{ID}(w_t) \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

We can approximate the unigram (UG), bi-gram (BG), and tri-gram (TG) features by using the observation vectors from the current frame only, from the previous and current frames (or the current and the next frames), and from the previous, current and next frames respectively as illustrated in Fig. 5. Li *et al.* [15] experimented using the true bi-gram and tri-grams features and compared them with the pseudo n-gram features we just described and did not observe significant difference in performance. Note that the feature construction approach used in our experiments as described above requires significantly fewer parameters to be estimated than alternative approaches. In fact, our approach requires only nV instead of V^n parameters to use n-gram features, where $n = \{1, 2, 3\}$ for the unigram, bi-gram,

TABLE II
SUMMARY OF THE WORD LABELING ACCURACY ON THE SEARCH QUERY TAGGING TASK USING UNIGRAM FEATURES

Model Settings		WLA (%)
Single-Layer Linear-Chain CRFs (UG)		87.5
Deep-Structured CRF (UG)	One layer (-T)	87.2
	Two layers (-T)	89.0
	Three layers (-T)	89.4
Best Published Results	Labeled only	88.7
	Labeled + unlabeled data	89.4
Upper Bound	Human labeling agreement	91.0

Only unigram features on the manually labeled portion of the training data were used in the single-layer linear-chain CRFs and the deep-structured CRFs settings. The best published results used unigram, bigram, lexicon, and regular expression features.

and tri-gram, respectively, and V is the vocabulary (lexicon) size. Although different types of n-gram features can be used, we conducted experiments using only the unigram and tri-gram features as we believe that insights on the model can be obtained using these two types of features. In all the experiments reported in this paper, the marginal posterior probabilities from the previous layer are treated differently than the original observation sequence, and additional features at jth layer are constructed on those posterior probabilities as $f_k^j(y_t, \boldsymbol{x}, t) = p(y_{t+k}^{j-1} \mid \boldsymbol{x^{j-1}})$, where $k \in [K_l^j, K_h^j]$ is the relative frame number in the $j-1$th layer and $K_h^j - K_l^j + 1$ is the total number of frames the jth layer depends on the $j-1$th layer. Note that these features can be constructed across different number of frames than the raw observation features do. For example, if unigram observation feature is used, only the current frame's observation is used in the first layer's feature construction. However, the system may use posterior probabilities in the previous and the next three-frames (i.e., $K_l^j = -3$ and $K_h^j = 3$) as additional features in the higher layers.

Table II summarizes the word labeling accuracy on the search query tagging task, where only unigram features on the labeled portion of the training data were used in both the single layer linear-chain CRF and the deep-structured CRF settings. No transition feature was used in the deep-structured CRF (indicated by $-$T in Table II) and so optimizing the sequence-level log-likelihood is equivalent to optimizing the frame-level log-likelihood.

In the second and third layers of the deep-structured CRF, the marginal posterior probabilities of the previous and next frames (words) were used as additional features to approximate the state dependency. The WLA result of 87.5% obtained by our single-layer linear-chain CRF is slightly better than that achieved by Li *et al.* [15] with the identical setting. As expected, the single-layer zeroth-order CRF without transition features slightly underperforms the single-layer linear-chain CRFs with WLAs of 87.2% and 87.5%, respectively. However, when two and three layers of the zeroth-order CRFs were used we have achieved WLAs of 89.0% and 89.4%, respectively. All the gains over the single-layer linear-chain CRF are statistically significant at significance level of 1%. In the experiments we also found that using the linear-chain CRF instead of zeroth-order CRF at the final layer does not improve the performance on this task. This is likely because each query in this task is very short with an average length of three words only.

The WLA of 89.4% is 0.7% better than the best published result of 88.7% on this task [15] obtained using the same labeled training set. The gain is statistically significant at significance level of 5%. Note that this best published result was achieved using unigram features, bigram features, and features extracted with the manually created regular expressions and field-dependent lexicons. In contrast, our result was obtained using the simple unigram features only. Our WLA result of 89.4% also matches the best published result obtained using both the labeled and unlabeled data sets.

In our additional experiments, we have also achieved a WLA of 89.5% using tri-gram features with only two layers of the zeroth-order CRF. However, since the best published results were obtained without using the tri-gram features, there is no simple comparison between this result and the best published result. Also note that the result of 89.5% is only 1.5% away from the human labeling agreement of 91% on this task.

B. *Task of Advertisement Field Segmentation*

As another natural language processing application, the goal of the advertisement field segmentation task is to classify each sentence in an advertisement into segments and assign a label to each segment. This task can also be considered as a sequential labeling problem where we provide a tag to each word in the sentences with words in the same segment assigned the same label. In our evaluation on this task, we also used the WLA as the evaluation metric to make our results comparable with the earlier work [4], [7], [16], [18] on the same task.

The data set we used for this task is the CLASSIFIEDS data provided by Grenager *et al.* [7] and consists of 8767 classified advertisements for apartment rentals in the San Francisco bay area. The data set was downloaded from the Craigslist website in June 2004. There are 12 predefined fields in this task, including size, rent, neighborhood, features, and so on. On average, each advertisement has 119 tokens segmented into 8.7 fields. This is very different from the search query tagging task in which the average query has only three words. Only 302 of the ads have been annotated. Following the earlier work [4], [7], [16], [18] on this task, the annotated ads are divided into a 102-ads training set, a 100-ads development set, and a 100-ads test set. The remaining 8465 ads form an unannotated training set which was used in the earlier work but not in our experiments.

The sentences were converted into word token sequences following exactly the same procedure adopted by all earlier work conducted on this task. Specifically, regular expressions were created to map phone numbers, email addresses, URLs, dates, money amounts and so on into special word tokens, and all other words and punctuations were directly mapped. However, we did not tokenize newline breaks, as was done in [4], which might be useful in determining sentence boundaries. Once we have the tokenized word sequences, we extract unigrams as features for CRFs and approximate the bi-gram and tri-gram features by using observation vectors from adjacent word tokens in the same way as that used in the search query tagging task.

TABLE III
SUMMARY OF THE WORD LABELING ACCURACY ON THE ADVERTISEMENT FIELD SEGMENTATION TASK

Model Settings		WLA (%) (UG)	WLA (%) (TG)
Deep-structured CRF S1	First layer (+T+S)	80.0	80.9
	Second layer (+T)	80.9	81.5
Deep-structured CRF S2	First layer (+T+F)	79.7	79.1
	Second layer (+T)	81.4	80.6
Deep-structured CRF S3	First layer (-T)	60.3	71.1
	Second layer (+T)	81.4	82.7
Best Published Results	Labeled data only	N/A	81.1
	Labeled + unlabeled data	N/A	82.9

Only the labeled portion of the training data was used in our experiments. *+T* and *-T* denote the conditions where transition features were used and not used, respectively; *+S* and *+F*, which are meaningful only to the first layer with transition features used, denote the conditions under which the sequence-level and frame-level optimization criteria were used, respectively. The second layer

Table III summaries the word labeling accuracy results on the advertisement field segmentation task, where only the labeled portion of the training data was used in our experiments. In Table III, results for three different deep-structured CRF settings are presented. In settings S1 and S2, transition features are used in both the first and second layers, with the first layer in S1 and S2 optimized using the sequence-level and frame-level criteria respectively. In setting S3, transition features are not used in the first layer but used in the second layer. The second layer is always optimized to maximize the sequence log-likelihood in all three settings. We do not show the third-layer results since the performance saturates at the third layers on this task likely due to the small training set.

Using the conventional single-layer linear-chain CRF, we see from Table III (layer one in setting S1) that we obtained 80.0% and 80.9% WLAs with the unigram and tri-gram features, respectively. In contrast, when transition features were not used (layer one in setting S3), only 60.3% (UG) and 71.1% (TG) WLAs were achieved using single-layer, zeroth-order CRFs. Although the first layer results were worse, 81.4% (UG) and 82.7% (TG) WLAs can be achieved when the second layer uses the transition features (layer two in setting S3). These are 0.5% and 1.2% better than the 80.9% (UG) and 81.5% (TG) WLAs obtained using two layers of the CRF with the transition features and sequence-level optimization criteria used in both layers (layer two in setting S1). These differences are statistically significant at significance level of 1%.

If the transition features and frame-level optimization criteria were used in the first layer the inference results become more complicated. When unigram features were used, the WLA achieved at the second layer of S2 is slightly (0.04%) better than that obtained using the two-layer CRF where no transition features were used in the first layer (layer two in setting S3) and is 0.5% better than that obtained with transition features used in both layers but with the first layer optimized using the sequence log-likelihood (layer two in S1). This indicates that optimizing the frame-level instead of sequence-level log-likelihood helps to improve the results at the final layer. However, if tri-gram features were used, using transition features at the first layer under-performs both the deep-structured CRF with no transition features used in the first layer (layer two in S3) and the deep-structured CRF with transition features used in both layers whose parameters were optimized using sequence-level likelihood (layer two in S1). This result puzzled us initially as we had expected to see better instead of worse results. After further analysis, we identified that it was caused by over-fitting at the first layer since there are only 102 ads in the training set and the number of model parameters is significantly increased when tri-gram features were used. The over-fitting at the first layer caused significant mismatch of the frame-level posterior probabilities passed into the second layer between the training and test sets. We have thus verified this by using half of the training set to train the first layer and the other half to train the second layer and achieved a comparable WLA as that obtained using the deep-structured CRFs without using transition features at the first layer (layer two in S3).

In all the experiments we have conducted and reported here, we used the posterior probabilities from both the previous and the next frames regardless of whether unigram features or tri-gram features were used. We have tried longer posterior probability dependencies and observed no significant difference on this task using the configurations listed in Table III. However, we did notice that when the transition feature is not used in the highest layer, increasing the range of posterior probability dependency helps a lot although the result is still not comparable with that achieved using the linear-chain CRF at the highest layer.

Our best WLA result of 82.7% is 1.6% better than the best published WLA result on this task [16] using the same tri-gram feature and labeled training set. This difference is statistically significant at significance level of 1%. This result is only 0.2% worse than the best result on this task using both labeled and unlabeled training data and using additional virtual evidences.

IV. CONCLUSION AND FUTURE WORK

We have developed and presented the deep-structured CRF model in this paper. We describe in detail the motivation, various architectures, and the parameter optimization criteria of this model. We have shown that the deep-structured CRF can achieve significant labeling accuracy improvement over the single-layer linear-chain CRF without significantly increasing the training and inference complexity in computation.

As demonstrated for the advertisement field segmentation task, optimizing the frame-level marginal probabilities in the lower layers of the deep-structured CRF model can achieve better labeling accuracy than optimizing the sequence-level probabilities if enough training data are available. However, it performs only slightly better than using the zeroth-order CRF in the lower layers even if over-fitting is not an issue. This suggests that we should use the zeroth-order CRF in the lower layers of the deep-structured CRF to improve the parameter estimation and state inference speed with slight scarification in the labeling accuracy.

We are improving the model in several aspects. First, no feature selection was conducted in this study. However, feature

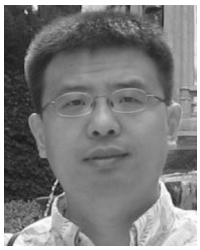
selection can be important to reduce the over-fitting problem. Second, we optimized the maximum conditional log-likelihood criterion at the highest layer. The criteria that are closer to the empirical error rate such as the minimum classification error and the maximum margin criteria may further improve the labeling accuracy. Third, we have assumed that the intermediate layers in the deep-structured CRF model have the same number of state values as in the final layer in this study. We have recently proposed techniques to infer the intermediate layers using discriminative criteria [30], [31]. Fourth, our model can be extended to incorporate the semi-supervised and unsupervised training criteria.

ACKNOWLEDGMENT

The authors would like to thank Dr. X. Li at Microsoft Research for valuable discussions and help in providing evaluation data sets in the experiments reported in this paper, Prof. G. Hinton at the University of Toronto for valuable discussions, and anonymous reviewers for constructive suggestions.

REFERENCES

[1] A. Arasu and H. Garcia-Molina, "Extracting structured data from webpage," in *Proc. ACM SIGMOD Int. Conf. Manage. of Data*, 2003.
[2] C. Barr, R. Jones, and M. Regelson, "The linguistic structure of English web-search queries," in *Proc. Conf. Empir. Meth. Natur. Lang. Process.*, 2008, pp. 1021–1030.
[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[4] M.-W. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," in *Proc. ACL*, 2007.
[5] W. W. Cohen and V. R. Carvalho, "Stacked sequential learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 671–676.
[6] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Ann. Math. Statist.*, vol. 43, pp. 1470–1480, 1972.
[7] L. Deng, D. Yu, and A. Acero, "A bidirectional target-filtering model of speech coarticulation and reduction: Two-stage implementation for phonetic recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 256–265, Jan. 2006.
[8] L. Deng, D. Yu, and A. Acero, "Structured speech modeling," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1492–1504, Sep. 2006.
[9] T. Grenager, D. Klein, and C. Manning, "Unsupervised learning of field segmentation models for information extraction," in *Proc. 43rd Annu. Meeting ACL*, 2005, pp. 371–378.
[10] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, vol. 3, pp. 1635–1638.
[11] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[13] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. ICCV*, 2009.
[14] X. Li, Y.-Y. Wang, and A. Acero, "Learning query intent from regularized click graph," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Jul. 2008.
[15] X. Li, Y.-Y. Wang, and A. Acero, "Extracting structured information from user queries with semi-supervised conditional random fields," in *Proc. SIGIR'09*, Jul. 2009.
[16] X. Li, "On the use of virtual evidence in conditional random fields," in *Proc. EMNLP*, Aug. 2009.
[17] L. Liao, D. Fox, and H. Kautz, "Hierarchical conditional random fields for GPS-based activity recognition," in *Proc. Int. Symp. Robot. Res. (ISRR)*, 2007.
[18] G. Mann and A. McCallum, "Generalized expectation criteria for semi-supervised learning of conditional random fields," in *Proc. ACL*, 2008.
[19] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Math. Comput.*, vol. 35, pp. 773–782, 1980.
[20] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003.
[21] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE ICNN*, 1993, vol. 1, pp. 586–591.
[22] C. Sutton, A. McCallum, and K. Rohanimanesh, "Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data," *J. Mach. Learn. Res.*, vol. 8, pp. 693–723, 2007.
[23] T. T. Truyen, "On Conditional Random Fields: Applications, Feature Selection, Parameter Estimation and Hierarchical Modelling," Ph.D. Dissertation, Curtin Univ. of Technol., Bentley, WA, Australia, 2008.
[24] P. Viola and M. Narasimhand, "Learning to extract information from semi-structured text using a discriminative context free grammar," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 330–337.
[25] D. Yu, L. Deng, and A. Acero, "Evaluation of a long-contextual-span hidden trajectory model and phonetic recognizer using A* lattice search," in *Proc. Interspeech*, 2005, pp. 553–556.
[26] D. Yu, L. Deng, and A. Acero, "A lattice search technique for a long-contextual-span hidden trajectory model of speech," *Speech Commun.*, vol. 48, no. 9, pp. 1214–1226, Sep. 2006.
[27] D. Yu, L. Deng, Y. Gong, and A. Acero, "A novel framework and training algorithm for variable-parameter hidden Markov models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1348–1360, Sep. 2009.
[28] D. Yu, L. Deng, and A. Acero, "Using continuous features in the maximum entropy model," *Pattern Recognition Lett.*, vol. 30, no. 8, Jun. 2009.
[29] D. Yu and L. Deng, "Solving nonlinear estimation problems using splines," *IEEE Signal Process. Mag.*, vol. 26, no. 4, pp. 86–90, Jul. 2009.
[30] D. Yu, L. Deng, and S. Wang, "Learning in the deep-structured conditional random fields," in *Proc. NIPS Workshop Deep Learn. Speech Recogn. Relat. Applicat.*, 2009.
[31] D. Yu, S. Wang, Z. Karam, and L. Deng, "Language recognition using deep-structured conditional random fields," in *Proc. ICASSP*, 2010, pp. 5030–5033.
[32] C. Zhao, J. Mahmud, and I. Ramakrishnan, "Exploiting structured reference data for unsupervised text segmentation with conditional random fields," in *Proc. SIAM Int. Conf. Data Mining*, 2008.
[33] J. Zhu, B. Zhang, Z. Nie, J.-R. Wen, and H.-W. Hon, "Webpage understanding: An integrated approach," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 903–912.



Dong Yu (M'97–SM'06) received the B.S. degree (with honor) in electrical engineering from Zhejiang University, Hangzhou, China, the M.S. degree in electrical engineering from the Chinese Academy of Sciences, Beijing, China, the M.S. degree in computer science from Indiana University, Bloomington, and the Ph.D. degree in computer science from the University of Idaho, Moscow.

He joined Microsoft Corporation in 1998 and the Microsoft Speech Research Group, Redmond, WA, in 2002, where he is a Researcher. His current research interests include speech processing, robust speech recognition, discriminative training, spoken dialog system, voice search technology, machine learning, and pattern recognition. He has published more than 60 papers in these areas and is the inventor/coinventor of more than 30 granted/pending patents.

Dr. Yu is currently serving as an Associate Editor of the IEEE SIGNAL PROCESSING MAGAZINE.

Shizhen Wang received the B.S. degree from Shandong University, Jinan, China, in 2002 and the M.S. degree from Tsinghua University, Beijing, China, in 2005, both in electrical engineering. He is currently working towards the Ph.D. degree in electrical engineering at University of California, Los Angeles (UCLA).

He is currently with Microsoft Corporation, Redmond, WA. His research interests include speech recognition, speaker normalization and adaptation, computer-aided language learning, and statistical signal processing.

Li Deng (M'86–SM'91–F'05) received the Ph.D. degree from the University of Wisconsin, Madison.

In 1989, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as an Assistant Professor, where he became a Full Professor in 1996. From 1992 to 1993, he conducted sabbatical research at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, and from 1997 to 1998, at ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In 1999, he joined Microsoft Research, Redmond, WA, as a Senior Researcher, where he is currently a Principal Researcher. He is also an Affiliate Professor in the Department of Electrical Engineering at the University of Washington, Seattle. His past and current research activities include automatic speech and speaker recognition, statistical methods and machine learning, neural information processing, machine intelligence, audio and acoustic signal processing, statistical signal processing and digital communication, human speech production and perception, acoustic phonetics, auditory speech processing, auditory physiology and modeling, noise robust speech processing, speech synthesis and enhancement, spoken language understanding systems, multimedia signal processing, and multimodal human–computer interaction. In these areas, he has published over 300 refereed papers in leading international conferences and journals, 12 book chapters, and has given keynotes, tutorials, and lectures worldwide. He has been granted over 30 U.S. or international patents in acoustics, speech/language technology, and signal processing. He authored or coauthored three books in speech processing and learning.

He serves on the Board of Governors of the IEEE Signal Processing Society and as Editor-in-Chief for the IEEE SIGNAL PROCESSING MAGAZINE. He is a Fellow of the Acoustical Society of America.