Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 3, September 2014, pp. 209-215
http://dx.doi.org/10.5391/IJFIS.2014.14.3.209

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Discriminative Training of Sequence Taggers via Local Feature Matching

Minyoung Kim
Department of Electronics and IT Media Engineering, Seoul National University of Science and Technology, Seoul 139-743, Korea

Abstract

Sequence tagging is the task of predicting frame-wise labels for a given input sequence and has important applications to diverse domains. Conventional methods such as maximum likelihood (ML) learning matches global features in empirical and model distributions, rather than local features, which directly translates into frame-wise prediction errors. Recent probabilistic sequence models such as conditional random fields (CRFs) have achieved great success in a variety of situations. In this paper, we introduce a novel discriminative CRF learning algorithm to minimize local feature mismatches. Unlike overall data fitting originating from global feature matching in ML learning, our approach reduces the total error over all frames in a sequence. We also provide an efficient gradient-based learning method via gradient forward-backward recursion, which requires the same computational complexity as ML learning. For several real-world sequence tagging problems, we empirically demonstrate that the proposed learning algorithm achieves significantly more accurate prediction performance than standard estimators.

Keywords: Machine learning, Sequence labeling/tagging, Conditional random fields

1. Introduction

Sequence tagging is the task of predicting frame-wise labels for a given input sequence. Often referred to as sequence segmentation, or structured output prediction depending on the domains of interest, sequence tagging has many important applications to diverse domains. For example, sequence tagging has been applied to annotating natural language sentences (parsing, part-of-speech tagging, named entity recognition), labeling biological sequences (protein secondary structure prediction), and estimating/tracking audio and/or video signals (speaker detection, body pose estimation of human motions).

The main difficulties of sequence tagging lie in the complex, chained, interdependent structure among predicted output variables. To effectively capture the output dependency, popular sequence models such as hidden Markov models (HMMs) have been adopted; however, this model is not optimal because of joint modeling of input and output data. The main concern is the output predictive performance. It is more attractive to focus on capturing the impact of inputs on outputs while circumventing the difficult modeling effort of input distributions [1, 2]. In light of these facts, the conditional random field (CRF) model emerged.

The goal of CRF is to represent the conditional distribution of an output sequence for a given input sequence. Its predictive power originates from its ability to capture the statistical

Received: Aug. 26, 2014
Revised : Sep. 19, 2014
Accepted: Sep. 20, 2014

Correspondence to: Minyoung Kim
(mikim@seoultech.ac.kr)

dependency of output variables via conditional probabilistic modeling. The CRF model has become one of the main computational tools for structured output prediction [3–9].

Standard maximum likelihood (ML) learning is the most popular estimator for training data in the CRF model. ML maximizes the conditional log-likelihood for the data when the optimization problem becomes convex. The optimality condition, or stationarity condition, is expressed as the matching of joint *global features* in empirical and model distributions. The global features are typically represented as the sum of local features over frames, where the local features capture the local dependency across input/output frames. Under the chain-structured dependency assumption, the local dependency is confined to the input/output variables of adjacent frames (current and next or previous and current frames).

The question arises as to whether the global feature matching of ML learning is indeed optimal for accurate label prediction. To answer this question, we suggest that local feature matching (LFM) (i.e., enforcing every local feature to be matched in empirical and model distributions) is more attractive for accurate output prediction since it directly translates into frame-wise prediction errors. We formulate a learning method as a local feature discrepancy minimization problem; specifically, the objective function is comprised of the sum of the squares of the norm differences between local features in empirical and model distributions. This is contrary to the strategy of ML learning, which forces the sum of the differences (instead of the squared norms) to vanish.

We present an efficient gradient-based learning method for the proposed formulation via gradient forward-backward recursion. In essence, the derivative of the recursion equation is calculated to obtain the output probabilistic inference of the CRF model. This gradient recursion is efficient as it requires the same computational complexity as ML learning. The proposed learning algorithm is tested on several real-world sequence tagging problems. We show that the proposed method significantly improves the prediction performance of standard estimators.

We begin by introducing the notation used in this paper and formalizing the statement of the problem to be solved in Section 2. We also provide a brief summary of the CRF model in this section. The main approach is described in Section 3, followed by empirical evaluations in Section 4. We conclude the paper in Section 5.

1.1 Preliminaries

The output sequence to be predicted is denoted by $\mathbf{Y} = y_1 \dots y_T$, where $y_t \in \{1, \dots, K\}$; this is known as K-way frame-wise classification. The output is predicted from the input sequence $\mathbf{X} = \mathbf{x}_1 \dots \mathbf{x}_T$, where $\mathbf{x}_t$ is a p-variate feature vector at time t. Note that the sequence length T is not fixed, but can vary from instance to instance. For example, for automatic speech recognition, the transcript sequence $\mathbf{Y}$ is predicted from the input sequence $\mathbf{X}$, which is comprised of certain acoustic features extracted from speech signals. Each tag y_t indicates the uttered word (out of K words) corresponding to the speech feature $\mathbf{x}_t$ at time t.

In this paper, we consider n available training data samples denoted by $\mathcal{D} = \{(\mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^n$. The sequence length of the i-th sample is denoted by T_i. Our goal is to learn the CRF model $P(\mathbf{Y}|\mathbf{X})$ from data and account for the complex statistical dependency among the input/output variables. In the next section, we briefly review the CRF model.

2. Conditional Random Fields

A CRF [3], denoted by $P(\mathbf{Y}|\mathbf{X})$, is a probabilistic model that represents the conditional distribution of a label sequence for a given input feature sequence. In particular, $P(\mathbf{Y}|\mathbf{X})$ can be written in exponential form as follows:

$$P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \frac{e^{F(\mathbf{X},\mathbf{Y};\boldsymbol{\theta})}}{Z(\mathbf{X};\boldsymbol{\theta})}, \quad Z(\mathbf{X};\boldsymbol{\theta}) = \sum_{\mathbf{Y}\in\mathcal{Y}} e^{F(\mathbf{X},\mathbf{Y};\boldsymbol{\theta})}, \quad (1)$$

where $\mathcal{Y} = \{1, \dots, K\}^T$ is the set of all possible output sequence realizations of sequence of length T, and $\boldsymbol{\theta}$ is the model's parameter vector. The denominator $Z(\mathbf{X};\boldsymbol{\theta})$, often called the partition function, makes (1) a distribution, which is dependent on input $\mathbf{X}$ and model parameter $\boldsymbol{\theta}$.

The core part of the model is the *feature function* $F(\mathbf{X}, \mathbf{Y};\boldsymbol{\theta})$, which represents the relationship between input and output variables. It is parametrized by $\boldsymbol{\theta}$, which affects the conditional distribution only through F. It is common to model F in linear form; that is,

$$F(\mathbf{X}, \mathbf{Y};\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \cdot \boldsymbol{\Psi}(\mathbf{X}, \mathbf{Y}), \quad (2)$$

where $\boldsymbol{\Psi}(\mathbf{X}, \mathbf{Y})$ is the input-output joint feature vector. Often referred to as the sufficient statistics of the CRF model, this label-combined observation effectively captures the interdependency between input and output variables.

The sequence data is temporally correlated, i.e., temporally nearby frames are mostly dependent on one another. Among other possible higher-order dynamics, first-order dependency modeling is the most popular for constructing joint features in CRF. In particular, the joint feature vector of $(\mathbf{X}, \mathbf{Y})$ is given by

$$\mathbf{\Psi}(\mathbf{X}, \mathbf{Y}) = \sum_{t=1}^{T} \boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1}). \tag{3}$$

This vector is determined by summing the *local features*

$$\boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1})$$

that are confined to capturing first-order dependency around frame t, where $t = 1, \ldots, T$. When $t = 1$, y_{t-1} is regarded as a void set, i.e., $\boldsymbol{\phi}_1(\mathbf{x}_1, y_1, y_0)$ implies $\boldsymbol{\phi}_1(\mathbf{x}_1, y_1)$.

The local features are often formed by label-indicated observation features. We define

$$\boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1}) = \Big[I(y_t = k \ \wedge \ y_{t-1} = l) \Big]_{K \times K} \otimes \mathbf{x}_t, \tag{4}$$

where $I(\cdot)$ is the delta function that returns 1 (0) if the argument is true (false), and $\otimes$ denotes the Kronecker product. Hence, $\boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1})$ is the $(K \times K \times p)$ tensor whose (k, l) block $((k, l) = 1, \ldots, K)$ is $\mathbf{x}_t$ if $y_t = k$ and $y_{t-1} = l$, and the $\mathbf{0}$ vector otherwise.

If the CRF model represents a distribution, probabilistic inference can be achieved. The most important inferences are $P(y_t|\mathbf{X})$ and $P(y_t, y_{t-1}|\mathbf{X})$. These quantities can be evaluated by forward/backward recursion, which is a type of dynamic programming equations. Now, we briefly describe the main results; for more details, please refer to [3, 5].

For the given input sequence $\mathbf{X}$, the potential function $M_t(\cdot)$ at frame t is defined by

$$\begin{aligned} M_t(y_t, y_{t-1}) &= e^{\boldsymbol{\theta}^\top \cdot \boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1})}, \ \ t \geq 2, \\ M_1(y_1) &= e^{\boldsymbol{\theta}^\top \cdot \boldsymbol{\phi}_1(\mathbf{x}_1, y_1)}. \end{aligned} \tag{5}$$

For a given $\mathbf{X}$, the potential functions for all t's can easily be evaluated. The resulting $M_t(y_t, y_{t-1})$ are usually stored in $(K \times K)$ matrices. Forward signals are then recursively defined. For the initial condition $\alpha_1(y_1) = M_1(y_1)$, and for $t = 2, \ldots, T$,

$$\alpha_t(y_t) = \sum_{y_{t-1}=1}^{K} \alpha_{t-1}(y_{t-1}) \cdot M_t(y_t, y_{t-1}). \tag{6}$$

These operations are essentially matrix-vector multiplication between M_t and the forward signal at the previous frame, $\alpha_{t-1}(y_{t-1})$.

The backward messages are similarly defined. For the initial condition $\beta_T(y_T) = 1$, and for $t < T$,

$$\beta_t(y_t) = \sum_{y_{t+1}=1}^{K} \beta_{t+1}(y_{t+1}) \cdot M_{t+1}(y_{t+1}, y_t). \tag{7}$$

It can be shown that

$$P(y_t|\mathbf{X}) \propto \alpha_t(y_t) \cdot \beta_t(y_t)$$

and

$$P(y_t, y_{t-1}|\mathbf{X}) \propto \alpha_{t-1}(y_{t-1}) \cdot M_t(y_t, y_{t-1}) \cdot \beta_t(y_t),$$

where the normalization constant in both cases is $Z(\mathbf{X})$. These computations are straightforward when the time complexity is proportional to the sequence length T.

For given data $\mathcal{D} = \{(\mathbf{X}^i, \mathbf{Y}^i)\}_{i=1}^n$, the conventional parameter estimator (ML) aims to maximize the total (conditional) likelihood:

$$\sum_{i=1}^{n} \log P(\mathbf{Y}^i|\mathbf{X}^i, \boldsymbol{\theta}) = \sum_{i=1}^{n} \Big(F(\mathbf{Y}^i|\mathbf{X}^i; \boldsymbol{\theta}) - \log Z(\mathbf{X}^i; \boldsymbol{\theta}) \Big). \tag{8}$$

Note that the objective function is concave in terms of $\boldsymbol{\theta}$ due to the linearity of F and convexity of $\log Z$. Hence, the optimal estimator satisfies the stationarity condition, i.e., the gradient vanishes. Setting the derivative to 0 yields (see [5]):

$$\sum_{i=1}^{n} \Big(\mathbf{\Psi}(\mathbf{X}^i, \mathbf{Y}^i) - \mathbb{E}\big[\mathbf{\Psi}(\mathbf{X}^i, \mathbf{Y}) \big] \Big) = 0, \tag{9}$$

where the expectation is taken with respect to $P(\mathbf{Y}|\mathbf{X}^i, \boldsymbol{\theta})$ for each i.

From Eq. (3), the expectation can easily be computed for a given $\boldsymbol{\theta}$ as follows: $\sum_{t=1}^{T} \mathbb{E}[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})]$, which only requires posterior distributions (from inference) $P(y_t, y_{t-1}|\mathbf{X})$. However, Eq. (9) is a complex nonlinear equation in terms of $\boldsymbol{\theta}$. Most CRF ML learning methods utilize gradient ascent.

3. CRF Training via LFM

Further investigation of Eq. (9) reveals that the ML estimator matches features summed over entire frames (i.e., *global features* as in Eq. (3)) between empirical and model expectations.

From the global-local feature relationship in Eq. (3), this results in

$$\sum_{i=1}^{n}\sum_{t=1}^{T_i}\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t^i, y_{t-1}^i) = \sum_{i=1}^{n}\sum_{t=1}^{T_i}\mathbb{E}\big[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})\big], \tag{10}$$

where the expectation is calculated with respect to $P(y_t, y_{t-1}|\mathbf{X}, \boldsymbol{\theta})$.

Thus, the ML estimator matches the empirical and model expected features in a global sense, i.e., matching the sums of local features. This is a weak constraint and is more intuitive (stronger condition) for matching individual local features as opposed to sums of them. This motivates our LFM training method that explicitly enforces

$$\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t^i, y_{t-1}^i) \approx \mathbb{E}\big[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})\big] \tag{11}$$

for each i and t. This idea can be formulated by minimizing the sum of squares of individual discrepancies. Specifically, we minimize

$$\sum_{i=1}^{n}\sum_{t=1}^{T_i}\Big\|\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t^i, y_{t-1}^i) - \mathbb{E}\big[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})\big]\Big\|^2. \tag{12}$$

Unlike the ML estimator, which forces the sum of local features matching, we explicitly enforce individual LFM using empirical and model expectations.

Our method is superior to the ML method in terms of its discriminative label prediction. In particular, both training methods minimize the disagreement between the expected model features and data; however, ML only considers each global feature (averaged over positions in a sequence), whereas our objective considers every sequence position. Clearly, the former is more related to overall data fitting, but our model significantly reduces the total error. We propose an efficient gradient descent optimization method using derivatives in forward/backward recursion.

3.1 Gradient Forward/Backward Recursion

Let $g_{i,t}(\boldsymbol{\theta}) = \mathbb{E}\big[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})\big] - \boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t^i, y_{t-1}^i)$. Then the objective in Eq. (12) to be minimized becomes

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{t=1}^{T_i}\|g_{i,t}(\boldsymbol{\theta})\|^2. \tag{13}$$

Gradient descent is performed by remapping $\boldsymbol{\theta}$ as follows: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$, where the step size η (> 0) is determined by line search. The gradient of the whole objective is summed over i and t for the gradients of $\|g_{i,t}(\boldsymbol{\theta})\|^2$, i.e., $2\frac{\partial g_{i,t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}g_{i,t}(\boldsymbol{\theta})$.

Evaluating $g_{i,t}(\boldsymbol{\theta})$ is straightforward, so instead we focus on deriving $\frac{\partial g_{i,t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$. Only the first term (expectation of local feature) of $g_{i,t}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$; that is,

$$\frac{\partial g_{i,t}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}}\mathbb{E}\big[\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1})\big] \tag{14}$$

$$= \frac{\partial}{\partial \boldsymbol{\theta}}\sum_{y_t, y_{t-1}} P(y_t, y_{t-1}|\mathbf{X}^i)\cdot\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1}) \tag{15}$$

$$= \sum_{y_t, y_{t-1}}\frac{\partial P(y_t, y_{t-1}|\mathbf{X}^i)}{\partial \boldsymbol{\theta}}\cdot\boldsymbol{\phi}_t(\mathbf{x}_t^i, y_t, y_{t-1}). \tag{16}$$

The key quantity is $\frac{\partial P(y_t, y_{t-1}|\mathbf{X}^i)}{\partial \boldsymbol{\theta}}$. Thus, we derive the log derivative $\frac{\partial \log P(y_t, y_{t-1}|\mathbf{X}^i)}{\partial \boldsymbol{\theta}}$ instead. It follows immediately that $\frac{\partial P}{\partial \boldsymbol{\theta}} = \frac{\partial \log P}{\partial \boldsymbol{\theta}}\cdot P$. Taking the gradient in the forward/backward equations yields

$$\frac{\partial \log P(y_t, y_{t-1}|\mathbf{X}^i)}{\partial \boldsymbol{\theta}} = \frac{\partial \log \alpha_{t-1}(y_{t-1})}{\partial \boldsymbol{\theta}} + \frac{\partial \log M_t(y_t, y_{t-1})}{\partial \boldsymbol{\theta}} + \frac{\partial \log \beta_t(y_t)}{\partial \boldsymbol{\theta}} - \frac{\partial \log Z(\mathbf{X}^i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{17}$$

The last term on the right-hand side is evaluated using conventional forward/backward recursion, and the other terms are obtained using the following derivative forward/backward procedure:

$$\frac{\partial \log M_t(y_t, y_{t-1})}{\partial \boldsymbol{\theta}} = \boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1}). \tag{18}$$

Note that for the remainder of the paper, the dependency on i is disregarded in the notation. Therefore, the log-forward/backward derivative recursion is

$$\frac{\partial \log \alpha_t(y_t)}{\partial \boldsymbol{\theta}} = \sum_{y_{t-1}} a_t(y_t, y_{t-1})\bigg(\frac{\partial \log \alpha_t(y_t)}{\partial \boldsymbol{\theta}} + \boldsymbol{\phi}_t(\mathbf{x}_t, y_t, y_{t-1})\bigg), \tag{19}$$

where $a_t(y_t, y_{t-1}) = \frac{\alpha_{t-1}(y_{t-1})\cdot M_t(y_t, y_{t-1})}{\alpha_t(y_t)}$. Furthermore,

$$\frac{\partial \log \beta_t(y_t)}{\partial \boldsymbol{\theta}} = \sum_{y_{t+1}} b_t(y_{t+1}, y_t)\bigg(\frac{\partial \log \beta_{t+1}(y_{t+1})}{\partial \boldsymbol{\theta}} + \boldsymbol{\phi}_{t+1}(\mathbf{x}_{t+1}, y_{t+1}, y_t)\bigg), \tag{20}$$

where $b_t(y_{t+1}, y_t) = \frac{\beta_{t+1}(y_{t+1}) \cdot M_{t+1}(y_{t+1}, y_t)}{\beta_t(y_t)}$.

Similar to an ML learning iteration, each gradient update requires a forward/backward pass through the entire sequence. Hence, the derivative forward/backward algorithm has the same complexity (linear in the sequence length) as the regular forward/backward algorithm. Since the proposed objective is non-convex, the choice of initial parameters is crucial. We chose to use the ML estimate as the initial iterate.

Often in CRF learning, additional regularization terms can be added to the objective to penalize large parameter values and to enforce a smooth model. We use the popular L2 norm, $\lambda||\boldsymbol{\theta}||^2$, for both the ML and proposed methods. Both methods are optimized by the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton gradient method [4].

4. Evaluation

In this section, we empirically evaluate the prediction performance of the proposed CRF learning algorithm based on LFM. The model is applied to several problems: speaking state detection, frequently-asked-question (FAQ) text document segmentation, and named entity tagging for natural language sentences.

Competing sequence tagging approaches are compared to the proposed method. These methods include the logistic regression method (`LogReg`), which treats individual frames as independent samples. More specifically, this model is trained frame-wise to predict y_t from $\mathbf{x}_t$; thus, the dependency structure that resides in the output labels is not accounted for. Another method is the HMM. Unlike conventional treatment, the state variables are observed in training data, and the state-conditional observation distribution $P(\mathbf{x}_t|y_t)$ is modeled as a Gaussian. This method is denoted by `GHMM`. Finally, the conventional ML CRF learning algorithm is denoted by `CRF-ML`, and our LFM approach is denoted by `CRF-LFM`.

4.1 Kiosk Speaking State Detection

The kiosk speaking state detection problem was considered. The goal of the task was to decide from a sequence of observations whether a human speaker is speaking ($y_t = 1$) or not ($y_t = 0$). Five labeled sequences of frames (length of approximately 2,000) were collected. The observation features at each time frame consisted of six binary features, three obtained from a face detector and the other three from audio cues. Four of the sequences were used for training and the remaining sequence was used for testing. For five different trials, the average test prediction errors were reported with standard deviations as shown in Table 1.

Table 1. Test prediction errors (%) for kiosk speaking state detection

Methods	Test errors
LogReg	6.94 ± 1.79
GHMM	5.48 ± 1.87
CRF-ML	4.88 ± 1.58
CRF-LFM	3.52 ± 1.33

GHMM, Gaussian hidden Markov model; CRF, conditional random field; ML, maximum likelihood; LFM, local feature matching.

The results indicated that the proposed LFM method considerably improves the prediction accuracy of the standard ML estimate. The logistic regression baseline method is inferior to the other sequence models because of its lack of temporal dependency. HMM, as expected, was inferior to the conditional CRF model due to its unnecessary effort in modeling the observation distribution.

4.2 FAQ Document Segmentation

The task of segmenting a FAQ document (a sequence of sentences) into four different types was considered: head, tail, question, and answer sentences. This is a type of information extraction problem that can be solved by sequence tagging methods.The FAQ document dataset in [10], comprised of approximately 40 documents obtained from seven Usenet multi-part FAQs, was used for the analysis. Each document can be viewed as a collection (sequence) of sentences, where the ordering of the document provides information relevant to type class labels.

As suggested in [10], 24 binary indicators were utilized as observation features, which mainly include word-level features; for instance, whether a sentence contains a question mark, alphanumerics, punctuation, special question words, or whether it is indented or not. The feature space was further enlarged by incorporating pairwise features; pairwise features were determined by applying the logical AND operator to every two features. Eighty percent of the data was used as training data, and 20% was used as test data; the simulations were randomly repeated five times. The prediction errors are summarized in Table 2.

Behaviors similar to that of the kiosk dataset were observed; again, the proposed method outperformed competing methods. In this case, the performance of HMM was not significantly better than that of the static logistic regression method. This can be explained by the fact that the temporal (ordered) dependency

Table 2. Test prediction errors (%) for FAQ document segmentation

Methods	Test errors
LogReg	16.71 ± 1.51
GHMM	15.82 ± 1.96
CRF-ML	12.71 ± 2.55
CRF-LFM	10.94 ± 2.36

GHMM, Gaussian hidden Markov model; CRF, conditional random field; ML, maximum likelihood; LFM, local feature matching.

Table 3. Test prediction errors (%) for named entity recognition

Methods	Test errors
LogReg	45.50
GHMM	30.48
CRF-ML	23.14
CRF-LFM	18.83

GHMM, Gaussian hidden Markov model; CRF, conditional random field; ML, maximum likelihood; LFM, local feature matching.

is weaker than that of the kiosk dataset. The CRF conditional modeling consistently worked better than the input modeling of HMM. Moreover, LFM yielded more accurate predictions than the ML estimator.

4.3 Named Entity Recognition

Next we tackled the named entity recognition problem in natural language processing. Given a sentence of words, each word is tagged as one of nine tags: B/I-person, B/I-organization, B/I-location, B/I-miscellaneous, and others. Here, B indicates that the word is a beginning tag while I means it is an intermediate word. Unlike the original nine-way classification setup, we considered a reduced tag set by not differentiating between beginning and intermediate tags.

Data from the CoNLL 2002 Spanish newswire corpus [11] was used. Our specific experimental setup was as follows: 2,000 randomly chosen sentences were selected from the dataset, which contains approximately $22,000$ different words. For the observation features of each word, word and spelling features were used (e.g., whether the word contains punctuators, numerics, or if the word length is greater than 3 characters). This formed just as many features as the number of different words. The data was randomly split into 1,000 training sentences and 1,000 test sentences. The prediction errors are shown in Table 3.

Behavior similar to the former two experiments was observed. The performance differences between logistic regression and the other sequence models indicated the importance of word correlation exploitation. Similar to the other datasets, the proposed LFM method exhibited stronger discriminative power than the existing approaches in output label tagging.

5. Conclusions

In this paper, we proposed a novel parameter learning method for CRFs to tackle the sequence tagging problem. Motivated by ML learning, we introduced a novel discriminative CRF learning algorithm to minimize local feature mismatches. While ML learning strives for overall data fitting, our approach reduces total error counts over all frames in a sequence. The proposed objective was optimized by gradient descent. Furthermore, we suggested efficient gradient forward/backward recursion equations. The superiority of the proposed method was demonstrated on several real-world sequence tagging problems. The proposed learning algorithm achieved significantly more accurate prediction performance than other standard estimators.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This study was supported by the Research Program funded by the Seoul National University of Science and Technology.

References

[1] A. Nadas, “A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 814–817, 1983. http://dx.doi.org/10.1109/TASSP.1983.1164173

[2] P. Woodland and D. Povey, “Large scale discriminative training for speech recognition,” in *Proceedings of ISCA Workshop on Automatic Speech Recognition: Challenges for the New Millennium*, Paris, France, 2000.

[3] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, June 28-July 1, 2001, pp. 282-289.

[4] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 134-141. http://dx.doi.org/10.3115/1073445.1073473

[5] S. Kumar and M. Hebert, "Discriminative random fields," *International Journal of Computer Vision*, vol. 68, no. 2, pp. 179-201, Jun. 2006. http://dx.doi.org/10.1007/s11263-006-7007-9

[6] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, December 13-18, 2004.

[7] R. McDonald and F. Pereira, "Identifying gene and protein mentions in text using conditional random fields," *BMC Bioinformatics*, vol. 6, no. Suppl 1, pp. S6, 2005. http://dx.doi.org/10.1186/1471-2105-6-S1-S6

[8] H. Xuming, R. S. Zemel, and M. A. Carreira-Perpindn, "Multiscale conditional random fields for image labeling," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, June 27-July 2, 2004, pp. II695-II702. http://dx.doi.org/10.1109/CVPR.2004.1315232

[9] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Proceedings of the 9th International Conference on Speech Communication and Technology*, Lisbon, Portugal, 2005.

[10] A. Mccallum, D. Freitag, and F. C. N. Pereira, "Maximum Entropy Markov Models for information extraction and segmentation," in *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA, 2000, pp. 591-598

[11] E. F. Tjong Kim Sang, "Introduction to the CoNLL-2002 shared task: language-independent named entity recognition," in *Proceedings of the 6th Conference on Natural Language Learning*, Taipei, Taiwn, 2002, pp. 155-158. http://dx.doi.org/10.3115/1118853.1118877

Minyoung Kim received his B.S. and M.S. degrees both in computer science and engineering from Seoul National University, South Korea. He earned a Ph.D. degree in computer science from Rutgers University in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an assistant professor in the Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.