Applying Conditional Random Fields to Japanese Morphological Analysis

Taku Kudo †* **Kaoru Yamamoto**‡ **Yuji Matsumoto** †
†Nara Institute of Science and Technology
8916-5, Takayama-Cho Ikoma, Nara, 630-0192 Japan
‡CREST JST, Tokyo Institute of Technology
4259, Nagatuta Midori-Ku Yokohama, 226-8503 Japan
taku-ku@is.naist.jp, kaoru@lr.pi.titech.ac.jp, matsu@is.naist.jp

Abstract

This paper presents Japanese morphological analysis based on conditional random fields (CRFs). Previous work in CRFs assumed that observation sequence (word) boundaries were fixed. However, word boundaries are not clear in Japanese, and hence a straightforward application of CRFs is not possible. We show how CRFs can be applied to situations where word boundary ambiguity exists. CRFs offer a solution to the long-standing problems in corpus-based or statistical Japanese morphological analysis. First, flexible feature designs for hierarchical tagsets become possible. Second, influences of label and length bias are minimized. We experiment CRFs on the standard testbed corpus used for Japanese morphological analysis, and evaluate our results using the same experimental dataset as the HMMs and MEMMs previously reported in this task. Our results confirm that CRFs not only solve the long-standing problems but also improve the performance over HMMs and MEMMs.

1 Introduction

Conditional random fields (CRFs) (Lafferty et al., 2001) applied to sequential labeling problems are conditional models, trained to discriminate the correct sequence from all other candidate sequences without making independence assumption for features. They are considered to be the state-of-the-art framework to date. Empirical successes with CRFs have been reported recently in part-of-speech tagging (Lafferty et al., 2001), shallow parsing (Sha and Pereira, 2003), named entity recognition (McCallum and Li, 2003), Chinese word segmentation (Peng et al., 2004), and Information Extraction (Pinto et al., 2003; Peng and McCallum, 2004).

Previous applications with CRFs assumed that observation sequence (e.g. word) boundaries are fixed, and the main focus was to predict label sequence (e.g. part-of-speech). However, word boundaries are not clear in non-segmented languages. One has to identify word segmentation as well as to predict part-of-speech in morphological analysis of non-segmented languages. In this paper, we show how CRFs can be applied to situations where word boundary ambiguity exists.

CRFs offer a solution to the problems in Japanese morphological analysis with hidden Markov models (HMMs) (e.g., (Asahara and Matsumoto, 2000)) or with maximum entropy Markov models (MEMMs) (e.g., (Uchimoto et al., 2001)). First, as HMMs are generative, it is hard to employ overlapping features stemmed from hierarchical tagsets and non-independent features of the inputs such as surrounding words, word suffixes and character types. These features have usually been ignored in HMMs, despite their effectiveness in unknown word guessing. Second, as mentioned in the literature, MEMMs could evade neither from *label bias* (Lafferty et al., 2001) nor from *length bias* (a bias occurring because of word boundary ambiguity). Easy sequences with low entropy are likely to be selected during decoding in MEMMs. The consequence is serious especially in Japanese morphological analysis due to hierarchical tagsets as well as word boundary ambiguity. The key advantage of CRFs is their flexibility to include a variety of features while avoiding these bias.

In what follows, we describe our motivations of applying CRFs to Japanese morphological analysis (Section 2). Then, CRFs and their parameter estimation are provided (Section 3). Finally, we discuss experimental results (Section 4) and give conclusions with possible future directions (Section 5).

2 Japanese Morphological Analysis

2.1 Word Boundary Ambiguity

Word boundary ambiguity cannot be ignored when dealing with non-segmented languages. A simple approach would be to let a character be a token (i.e., character-based Begin/Inside tagging) so that boundary ambiguity never occur (Peng et al., 2004).

*At present, NTT Communication Science Laboratories, 2-4, Hikaridai, Seika-cho, Soraku, Kyoto, 619-0237 Japan taku@cslab.kecl.ntt.co.jp

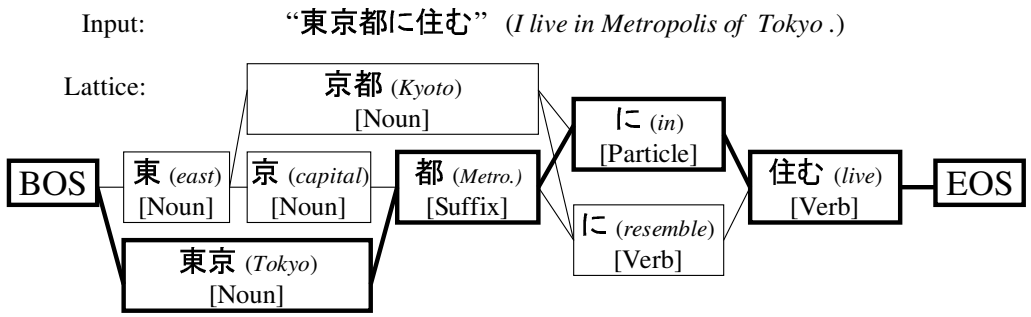

Figure 1: Example of lattice for Japanese morphological analysis

However, B/I tagging is not a standard method in 20-year history of corpus-based Japanese morphological analysis. This is because B/I tagging cannot directly reflect lexicons which contain prior knowledge about word segmentation. We cannot ignore a lexicon since over 90% accuracy can be achieved even using the longest prefix matching with the lexicon. Moreover, B/I tagging produces a number of redundant candidates which makes the decoding speed slower.

Traditionally in Japanese morphological analysis, we assume that a lexicon, which lists a pair of a word and its corresponding part-of-speech, is available. The lexicon gives a tractable way to build a lattice from an input sentence. A lattice represents all candidate paths or all candidate sequences of tokens, where each token denotes a word with its part-of-speech [1].

Figure 1 shows an example where a total of 6 candidate paths are encoded and the optimal path is marked with bold type. As we see, the set of labels to predict and the set of states in the lattice are different, unlike English part-of-speech tagging that word boundary ambiguity does not exist.

Formally, the task of Japanese morphological analysis can be defined as follows. Let $\mathbf{x}$ be an input, unsegmented sentence. Let $\mathbf{y}$ be a path, a sequence of tokens where each token is a pair of word w_i and its part-of-speech t_i. In other words, $\mathbf{y} = (\langle w_1, t_1 \rangle, \ldots, \langle w_{\#\mathbf{y}}, t_{\#\mathbf{y}} \rangle)$ where $\#\mathbf{y}$ is the number of tokens in the path $\mathbf{y}$. Let $\mathcal{Y}(\mathbf{x})$ be a set of candidate paths in a lattice built from the input sentence $\mathbf{x}$ and a lexicon. The goal is to select a correct path $\hat{\mathbf{y}}$ from all candidate paths in the $\mathcal{Y}(\mathbf{x})$. The distinct property of Japanese morphological analysis is that the number of tokens $\mathbf{y}$ varies, since the set of labels and the set of states are not the same.

[1] If one cannot build a lattice because no matching word can be found in the lexicon, unknown word processing is invoked. Here, candidate tokens are built using character types, such as *hiragana*, *katakana*, Chinese characters, alphabets, and numbers.

2.2 Long-Standing Problems

2.2.1 Hierarchical Tagset

Japanese part-of-speech (POS) tagsets used in the two major Japanese morphological analyzers ChaSen[2] and JUMAN[3] take the form of a hierarchical structure. For example, IPA tagset[4] used in ChaSen consists of three categories: part-of-speech, conjugation form (cform), and conjugate type (ctype). The cform and ctype are assigned only to words that conjugate, such as verbs and adjectives. The part-of-speech has at most four levels of subcategories. The top level has 15 different categories, such as *Noun*, *Verb*, etc. *Noun* is subdivided into *Common Noun*, *Proper Noun* and so on. *Proper Noun* is again subdivided into *Person*, *Organization* or *Place*, etc. The bottom level can be thought as the *word level* (base form) with which we can completely discriminate all words as different POS. If we distinguish each branch of the hierarchical tree as a different label (ignoring the word level), the total number amounts to about 500, which is much larger than the typical English POS tagset such as Penn Treebank.

The major effort has been devoted how to interpolate each level of the hierarchical structure as well as to exploit atomic features such as word suffixes and character types. If we only use the bottom level, we suffer from the data sparseness problem. On the other hand, if we use the top level, we lack in granularity of POS to capture fine differences. For instance, some suffixes (e.g., *san* or *kun*) appear after names, and are helpful to detect words with *Name* POS. In addition, the conjugation form (cfrom) must be distinguished appearing only in the succeeding position in a bi-gram, since it is dominated by the word appearing in the next.

Asahara et al. extended HMMs so as to incorporate 1) position-wise grouping, 2) word-level statis-

[2] http://chasen.naist.jp/

[3] http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman.html

[4] http://chasen.naist.jp/stable/ipadic/

tics, and 3) smoothing of word and POS level statistics (Asahara and Matsumoto, 2000). However, the proposed method failed to capture non-independent features such as suffixes and character types and selected smoothing parameters in an ad-hoc way.

2.2.2 Label Bias and Length Bias

It is known that maximum entropy Markov models (MEMMs) (McCallum et al., 2000) or other discriminative models with independently trained next-state classifiers potentially suffer from the *label bias* (Lafferty et al., 2001) and *length bias*. In Japanese morphological analysis, they are extremely serious problems. This is because, as shown in Figure 1, the *branching variance* is considerably high, and the number of tokens varies according to the output path.

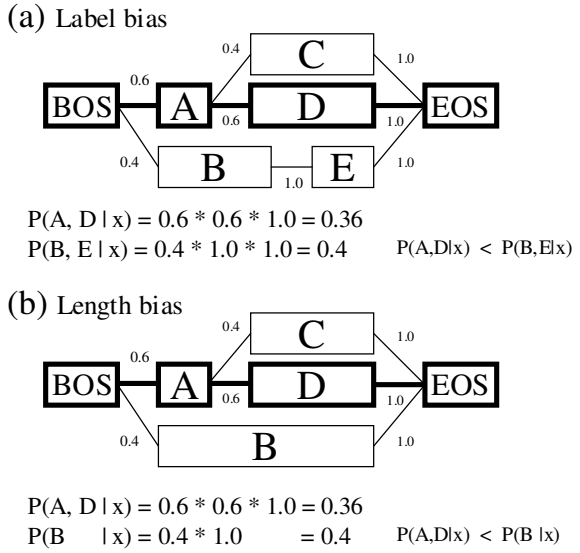

Figure 2: Label and length bias in a lattice

An example of the *label bias* is illustrated in Figure 2:(a) where the path is searched by sequential combinations of maximum entropy models (MEMMs), i.e., $P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{\#\mathbf{y}} p(\langle w_i, t_i\rangle|\langle w_{i-1}, t_{i-1}\rangle)$. Even if MEMMs learn the correct path A-D with independently trained maximum entropy models, the path B-E will have a higher probability and then be selected in decoding. This is because the token B has only the single outgoing token E, and the transition probability for B-E is always 1.0. Generally speaking, the complexities of transitions vary according to the tokens, and the transition probabilities with low-entropy will be estimated high in decoding. This problem occurs because the training is performed only using the correct path, ignoring all other transitions.

Moreover, we cannot ignore the influence of the *length bias* either. By the length bias, we mean that short paths, consisting of a small number of tokens, are preferred to long path. Even if the transition probability of each token is small, the total probability of the path will be amplified when the path is short 2:(b)). Length bias occurs in Japanese morphological analysis because the number of output tokens $\mathbf{y}$ varies by use of prior lexicons.

Uchimoto et al. attempted a variant of MEMMs for Japanese morphological analysis with a number of features including suffixes and character types (Uchimoto et al., 2001; Uchimoto et al., 2002; Uchimoto et al., 2003). Although the performance of unknown words were improved, that of known words degraded due to the label and length bias. Wrong segmentation had been reported in sentences which are analyzed correctly by naive rule-based or HMMs-based analyzers.

3 Conditional Random Fields

Conditional random fields (CRFs) (Lafferty et al., 2001) overcome the problems described in Section 2.2. CRFs are discriminative models and can thus capture many correlated features of the inputs. This allows flexible feature designs for hierarchical tagsets. CRFs have a single exponential model for the joint probability of the entire paths given the input sentence, while MEMMs consist of a sequential combination of exponential models, each of which estimates a conditional probability of next tokens given the current state. This minimizes the influences of the label and length bias.

As explained in Section 2.1, there is word boundary ambiguity in Japanese, and we choose to use a lattice instead of B/I tagging. This implies that the set of labels and the set of states are different, and the number of tokens $\#\mathbf{y}$ varies according to a path. In order to accomodate this, we define CRFs for Japanese morphological analysis as the conditional probability of an output path $\mathbf{y} = (\langle w_1, t_1\rangle, \ldots, \langle w_{\#\mathbf{y}}, t_{\#\mathbf{y}}\rangle)$ given an input sequence $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\Big(\sum_{i=1}^{\#\mathbf{y}} \sum_{k} \lambda_k f_k(\langle w_{i-1}, t_{i-1}\rangle, \langle w_i, t_i\rangle)\Big),$$

where $Z_{\mathbf{x}}$ is a normalization factor over all candidate paths, i.e.,

$$Z_{\mathbf{x}} = \sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \exp\Big(\sum_{i=1}^{\#\mathbf{y}'} \sum_{k} \lambda_k f_k(\langle w'_{i-1}, t'_{i-1}\rangle, \langle w'_i, t'_i\rangle)\Big),$$

$f_k(\langle w_{i-1}, t_{i-1}\rangle, \langle w_i, t_i\rangle)$ is an arbitrary feature function over i-th token $\langle w_i, t_i\rangle$, and its previous token $\langle w_{i-1}, t_{i-1}\rangle$ [5]. $\lambda_k (\in \mathbf{\Lambda} = \{\lambda_1, \ldots, \lambda_K\} \in \mathbb{R}^K)$ is a learned weight or parameter associated with feature function f_k.

Note that our formulation of CRFs is different from the widely-used formulations (e.g., (Sha and Pereira, 2003; McCallum and Li, 2003; Peng et al., 2004; Pinto et al., 2003; Peng and McCallum, 2004)). The previous applications of CRFs assign a conditional probability for a label sequence $\mathbf{y} = y_1, \ldots, y_T$ given an input sequence $\mathbf{x} = x_1, \ldots, x_T$ as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\Big(\sum_{i=1}^{T} \sum_k \lambda_k f_k(y_{i-1}, y_i, \mathbf{x})\Big)$$

In our formulation, CRFs deal with word boundary ambiguity. Thus, the the size of output sequence T is not fixed through all candidates $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$. The index i is not tied with the input $\mathbf{x}$ as in the original CRFs, but unique to the output $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$.

Here, we introduce the *global feature vector* $\mathbf{F}(\mathbf{y}, \mathbf{x}) = \{F_1(\mathbf{y}, \mathbf{x}), \ldots, F_K(\mathbf{y}, \mathbf{x})\}$, where $F_k(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{\#\mathbf{y}} f_k(\langle w_{i-1}, t_{i-1}\rangle, \langle w_i, t_i\rangle)$. Using the global feature vector, $P(\mathbf{y}|\mathbf{x})$ can also be represented as $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp(\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}))$. The most probable path $\hat{\mathbf{y}}$ for the input sentence $\mathbf{x}$ is then given by

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} P(\mathbf{y}|\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}),$$

which can be found with the Viterbi algorithm. An interesting note is that the decoding process of CRFs can be reduced into a simple linear combinations over all global features.

3.1 Parameter Estimation

CRFs are trained using the standard maximum likelihood estimation, i.e., maximizing the log-likelihood $\mathcal{L}_\mathbf{\Lambda}$ of a given training set $T = \{\langle \mathbf{x}_j, \mathbf{y}_j\rangle\}_{j=1}^N$,

$$\begin{aligned}
\hat{\mathbf{\Lambda}} &= \operatorname*{argmax}_{\mathbf{\Lambda} \in \mathbb{R}^K} \mathcal{L}_\mathbf{\Lambda}, \quad where \\
\mathcal{L}_\mathbf{\Lambda} &= \sum_j \log(P(\mathbf{y}_j|\mathbf{x}_j)) \\
&= \sum_j \Big[\log\Big(\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)} \exp\left(\mathbf{\Lambda} \cdot [\mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \mathbf{F}(\mathbf{y}, \mathbf{x}_j)]\right)\Big)\Big] \\
&= \sum_j \Big[\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \log(Z_{\mathbf{x}_j})\Big].
\end{aligned}$$

[5] We could use trigram or more general n-gram feature functions (e.g., $f_k(\langle w_{i-n}, t_{i-n}\rangle, \ldots, \langle w_i, t_i\rangle)$), however we restrict ourselves to bi-gram features for clarity.

To maximize $L_\mathbf{\Lambda}$, we have to maximize the difference between the inner product (or *score*) of the correct path $\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j)$ and those of all other candidates $\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}_j)$, $\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)$. CRFs is thus trained to discriminate the correct path from all other candidates, which reduces the influences of the label and length bias in encoding.

At the optimal point, the first-derivative of the log-likelihood becomes 0, thus,

$$\begin{aligned}
\frac{\delta \mathcal{L}_\mathbf{\Lambda}}{\delta \lambda_k} &= \sum_j \Big(F_k(\mathbf{y}_j, \mathbf{x}_j) - E_{P(\mathbf{y}|\mathbf{x}_j)}\big[F_k(\mathbf{y}, \mathbf{x}_j)\big]\Big) \\
&= O_k - E_k = 0,
\end{aligned}$$

where $O_k = \sum_j F_k(\mathbf{y}_j, \mathbf{x}_j)$ is the count of feature k observed in the training data T, and $E_k = \sum_j E_{P(\mathbf{y}|\mathbf{x}_j)}[F_k(\mathbf{y}, \mathbf{x}_j)]$ is the expectation of feature k over the model distribution $P(\mathbf{y}|\mathbf{x})$ and T. The expectation can efficiently be calculated using a variant of the forward-backward algorithm.

$$E_{P(\mathbf{y}|\mathbf{x})}[F_k(\mathbf{y}, \mathbf{x})] = \sum_{\{\langle w', t'\rangle, \langle w, t\rangle\} \in \mathcal{B}(\mathbf{x})} \frac{\alpha_{\langle w', t'\rangle} \cdot f_k^* \cdot \exp(\sum_{k'} \lambda_{k'} f_{k'}^*) \cdot \beta_{\langle w, t\rangle}}{Z_\mathbf{x}},$$

where f_k^* is an abbreviation for $f_k(\langle w', t'\rangle, \langle w, t\rangle)$, $\mathcal{B}(\mathbf{x})$ is a set of all bi-gram sequences observed in the lattice for $\mathbf{x}$, and $\alpha_{\langle w,t\rangle}$ and $\beta_{\langle w,t\rangle}$ are the forward-backward costs given by the following recursive definitions:

$$\begin{aligned}
\alpha_{\langle w,t\rangle} &= \sum_{\langle w', t'\rangle \in LT(\langle w,t\rangle)} \alpha_{\langle w', t'\rangle} \cdot \exp\Big(\sum_k \lambda_k f_k(\langle w', t'\rangle, \langle w, t\rangle)\Big) \\
\beta_{\langle w,t\rangle} &= \sum_{\langle w', t'\rangle \in RT(\langle w,t\rangle)} \beta_{\langle w', t'\rangle} \cdot \exp\Big(\sum_k \lambda_k f_k(\langle w, t\rangle, \langle w', t'\rangle)\Big),
\end{aligned}$$

where $LT(\langle w, t\rangle)$ and $RT(\langle w, t\rangle)$ denote a set of tokens each of which connects to the token $\langle w, t\rangle$ from the left and the right respectively. Note that initial costs of two virtual tokens, $\alpha_{\langle w_{bos}, t_{bos}\rangle}$ and $\beta_{\langle w_{eos}, t_{eos}\rangle}$, are set to be 1. A normalization constant is then given by $Z_\mathbf{x} = \alpha_{\langle w_{eos}, t_{eos}\rangle} (= \beta_{\langle w_{bos}, t_{bos}\rangle})$.

We attempt two types of regularizations in order to avoid overfitting. They are a Gaussian prior (L2-norm) (Chen and Rosenfeld, 1999) and a Laplacian prior (L1-norm) (Goodman, 2004; Peng and McCallum, 2004)

$$\mathcal{L}_\mathbf{\Lambda} = C \sum_j \log(P(\mathbf{y}_j|\mathbf{x}_j)) - \frac{1}{2} \begin{cases} \sum_k |\lambda_k| & \text{(L1-norm)} \\ \sum_k |\lambda_k|^2 & \text{(L2-norm)} \end{cases}$$

Below, we refer to CRFs with L1-norm and L2-norm regularization as L1-CRFs and L2-CRFs respectively. The parameter $C \in \mathbb{R}^+$ is a hyperparameter of CRFs determined by a cross validation.

L1-CRFs can be reformulated into the constrained optimization problem below by letting $\lambda_k = \lambda_k^+ - \lambda_k^-$:

$$\begin{aligned} \max: \quad & C \sum_j \log(P(\mathbf{y}_j|\mathbf{x}_j)) - \sum_k (\lambda_k^+ + \lambda_k^-)/2 \\ \text{s.t.}, \quad & \lambda_k^+ \geq 0,\ \lambda_k^- \geq 0. \end{aligned}$$

At the optimal point, the following Karush-Kuhun-Tucker conditions satisfy: $\lambda_k^+ \cdot [C \cdot (O_k - E_k) - 1/2] = 0, \lambda_k^- \cdot [C \cdot (E_k - O_k) - 1/2] = 0$, and $|C \cdot (O_k - E_k)| \leq 1/2$. These conditions mean that both λ_k^+ and λ_k^- are set to be 0 (i.e., $\lambda_k = 0$), when $|C \cdot (O_k - E_k)| < 1/2$. A non-zero weight is assigned to λ_k, only when $|C \cdot (O_k - E_k)| = 1/2$. L2-CRFs, in contrast, give the optimal solution when $\frac{\delta \mathcal{L}_{\mathbf{\Lambda}}}{\delta \lambda_k} = C \cdot (O_k - E_k) - \lambda_k = 0$. Omitting the proof, $(O_k - E_k) \neq 0$ can be shown and L2-CRFs thus give a non-sparse solution where all λ_k have non-zero weights.

The relationship between two reguralizations have been studied in Machine Learning community. (Perkins et al., 2003) reported that L1-regularizer should be chosen for a problem where most of given features are *irrelevant*. On the other hand, L2-regularizer should be chosen when most of given features are *relevant*. An advantage of L1-based regularizer is that it often leads to sparse solutions where most of λ_k are exactly 0. The features assigned zero weight are thought as *irrelevant* features to classifications. The L2-based regularizer, also seen in SVMs, produces a non-sparse solution where all of λ_k have non-zero weights. All features are used with L2-CRFs.

The optimal solutions of L2-CRFs can be obtained by using traditional iterative scaling algorithms (e.g., IIS or GIS (Pietra et al., 1997)) or more efficient quasi-Newton methods (e.g., L-BFGS (Liu and Nocedal, 1989)). For L1-CRFs, constrained optimizers (e.g., L-BFGS-B (Byrd et al., 1995)) can be used.

4 Experiments and Discussion

4.1 Experimental Settings

We use two widely-used Japanese annotated corpora in the research community, Kyoto University Corpus ver 2.0 (**KC**) and RWCP Text Corpus (**RWCP**), for our experiments on CRFs. Note that each corpus has a different POS tagset and details (e.g., size of training and test dataset) are summarized in Table 1.

One of the advantages of CRFs is that they are flexible enough to capture many correlated features, including overlapping and non-independent features. We thus use as many features as possible, which could not be used in HMMs. Table 2 summarizes the set of feature templates used in the **KC** data. The templates for **RWCP** are essentially the same as those of **KC** except for the maximum level of POS subcatgeories. Word-level templates are employed when the words are *lexicalized*, i.e., those that belong to particle, auxiliary verb, or suffix[6]. For an unknown word, length of the word, up to 2 suffixes/prefixes and character types are used as the features. We use all features observed in the lattice without any cut-off thresholds. Table 1 also includes the number of features in both data sets.

We evaluate performance with the standard F-score ($F_{\beta=1}$) defined as follows:

$$\begin{aligned} F_{\beta=1} &= \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}, \\ \text{where} \quad Recall &= \frac{\#\text{ of correct tokens}}{\#\text{ of tokens in test corpus}} \\ Precision &= \frac{\#\text{ of correct tokens}}{\#\text{ of tokens in system output}}. \end{aligned}$$

In the evaluations of F-scores, three criteria of correctness are used: *seg*: (only the word segmentation is evaluated), *top*: (word segmentation and the top level of POS are evaluated), and *all*: (all information is used for evaluation).

The hyperparameters C for L1-CRFs and L2-CRFs are selected by cross-validation. Experiments are implemented in C++ and executed on Linux with XEON 2.8 GHz dual processors and 4.0 Gbyte of main memory.

4.2 Results

Tables 3 and 4 show experimental results using **KC** and **RWCP** respectively. The three F-scores (*seg/top/all*) for our CRFs and a baseline bi-gram HMMs are listed.

In Table 3 (**KC** data set), the results of a variant of maximum entropy Markov models (MEMMs) (Uchimoto et al., 2001) and a rule-based analyzer (JUMAN[7]) are also shown. To make a fare comparison, we use exactly the same data as (Uchimoto et al., 2001).

In Table 4 (**RWCP** data set), the result of an extended Hidden Markov Models (E-HMMs) (Asa-

[6] These lexicalizations are usually employed in Japanese morphological analysis.

[7] JUMAN assigns "unknown POS" to the words not seen in the lexicon. We simply replace the POS of these words with the default POS, *Noun-SAHEN*.

Table 1: Details of Data Set

	KC	**RWCP**
source	Mainich News Article ('95)	Mainich News Article ('94)
lexicon (# of words)	JUMAN ver. 3.61 (1,983,173)	IPADIC ver. 2.7.0 (379,010)
POS structure	2-levels POS, cfrom, ctype, base form	4-levels POS, cfrom, ctype, base form
# of training sentences	7,958 (Articles on Jan. 1st - Jan. 8th)	10,000 (first 10,000 sentences)
# of training tokens	198,514	265,631
# of test sentences	1,246 (Articles on Jan. 9th)	25,743 (all remaining sentences)
# of test tokens	31,302	655,710
# of features	791,798	580,032

Table 2: Feature templates: $f_k(\langle w', t' \rangle, \langle w, t \rangle)$
$t' = \langle p1', p2', cf', ct, bw' \rangle$, $t = \langle p1, p2, cf, ct, bw \rangle$, where $p1'/p1$ and $p2'/p2$ are the top and sub categories of POS. cf'/cf and ct'/ct are the cfrom and ctype respectively. bw'/bw are the base form of the words w'/w.

type	template
Unigram basic features	$\langle p1 \rangle$
	$\langle p1, p2 \rangle$
w is known	$\langle bw \rangle$
	$\langle bw, p1 \rangle$
	$\langle bw, p1, p2 \rangle$
w is unknown	length of the word w
	up to 2 suffixes $\times$ $\{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$
	up to 2 prefixes $\times$ $\{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$
	character type $\times$ $\{\phi, \langle p1 \rangle, \langle p1, p2 \rangle\}$
Bigram basic features	$\langle p1', p1 \rangle$
	$\langle p1', p1, p2 \rangle$
	$\langle p1', p2', p1 \rangle$
	$\langle p1', p2', p1, p2 \rangle$
	$\langle p1', p2', cf', p1, p2 \rangle$
	$\langle p1', p2', ct', p1, p2 \rangle$
	$\langle p1', p2', cf', ct', p1, p2 \rangle$
	$\langle p1', p2', p1, p2, cf \rangle$
	$\langle p1', p2', p1, p2, ct \rangle$
	$\langle p1', p2', p1, p2, cf, ct \rangle$
	$\langle p1', p2', cf', p1, p2, cf \rangle$
	$\langle p1', p2', ct, p1, p2, ct \rangle$
	$\langle p1', p2', cf', p1, p2, ct \rangle$
	$\langle p1', p2', ct', p1, p2, cf \rangle$
	$\langle p1', p2', cf', ct', p1, p2, cf, ct \rangle$
w' is lexicalized	$\langle p1', p2', cf', ct', bw', p1, p2 \rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, cf \rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, ct \rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, cf, ct \rangle$
w is lexicalized	$\langle p1', p2', p1, p2, cf, ct, bw \rangle$
	$\langle p1', p2', cf', p1, p2, cf, ct, bw \rangle$
	$\langle p1', p2', ct', p1, p2, cf, ct, bw \rangle$
	$\langle p1', p2', cf', ct', p1, p2, cf, ct, bw \rangle$
w'/w are lexicalized	$\langle p1', p2', cf', ct', bw', p1, p2, cf, ct, bw \rangle$

hara and Matsumoto, 2000) trained and tested with the same corpus is also shown. E-HMMs is applied to the current implementation of ChaSen. Details of E-HMMs are described in Section 4.3.2.

We directly evaluated the difference of these systems using McNemar's test. Since there are no standard methods to evaluate the significance of F scores, we convert the outputs into the character-based B/I labels and then employ a McNemar's paired test on the labeling disagreements. This evaluation was also used in (Sha and Pereira, 2003). The results of McNemar's test suggest that L2-CRFs is significantly better than other systems including L1-CRFs[8]. The overall results support our empirical success of morphological analysis based on CRFs.

4.3 Discussion

4.3.1 CRFs and MEMMs

Uchimoto el al. proposed a variant of MEMMs trained with a number of features (Uchimoto et al., 2001). Although they improved the accuracy for unknown words, they fail to segment some sentences which are correctly segmented with HMMs or rule-based analyzers.

Figure 3 illustrates the sentences which are incorrectly segmented by Uchimoto's MEMMs. The correct paths are indicated by bold boxes. Uchimoto et al. concluded that these errors were caused by non-standard entries in the lexicon. In Figure 3, "ロマンは" *(romanticist)* and "ない心" *(one's heart)* are unusual spellings and they are normally written as "ロマン派" and "内心" respectively. However, we conjecture that these errors are caused by the influence of the length bias. To support our claim, these sentences are correctly segmented by CRFs, HMMs and rule-based analyzers using the same lexicon as (Uchimoto et al., 2001). By the length bias, short paths are preferred to long paths. Thus, single token "ロマンは" or "ない心" is likely to be selected compared to multiple tokens "ロマン / は" or "ない / 心". Moreover, "ロマン" and "ロマンは" have exactly the same POS (*Noun*), and transition probabilities of these tokens become almost equal. Consequentially, there is no choice but to select a short path (single token) in order to maximize the whole sentence probability.

Table 5 summarizes the number of errors in HMMs, CRFs and MEMMs, using the KC data set. Two types of errors, *l*-error and *s*-error, are given in

[8] In all cases, the p-values are less than 1.0×10^{-4}.

Table 3: Results of KC, ($F_{\beta=1}$ (precision/recall))

system	seg	top	all
L2-CRFs ($C=1.2$)	**98.96 (99.04/98.88)**	**98.31 (98.39/98.22)**	**96.75 (96.83/96.67)**
L1-CRFs ($C=3.0$)	98.80 (98.84/98.77)	98.14 (98.18/98.11)	96.55 (96.58/96.51)
MEMMs (Uchimoto 01)	96.44 (95.78/97.10)	95.81 (95.15/96.47)	94.27 (93.62/94.92)
JUMAN (rule-based)	98.70 (98.88/98.51)	98.09 (98.27/97.91)	93.73 (93.91/93.56)
HMMs-bigram (baseline)	96.22 (96.16/96.28)	94.96 (94.90/95.02)	91.85 (91.79/91.90)

Table 4: Results of RWCP, ($F_{\beta=1}$ (precision/recall))

system	seg	top	all
L2-CRFs ($C=2.4$)	**99.11 (99.03/99.20)**	**98.73 (98.65/98.81)**	**97.66 (97.58/97.75)**
L1-CRFs ($C=3.0$)	99.00 (98.86/99.13)	98.58 (98.44/98.72)	97.30 (97.16/97.43)
E-HMMs (Asahara 00)	98.87 (98.77/98.97)	98.33 (98.23/98.43)	96.95 (96.85/97.04)
HMMs-bigram (baseline)	98.82 (98.69/98.94)	98.10 (97.97/98.22)	95.90 (95.78/96.03)

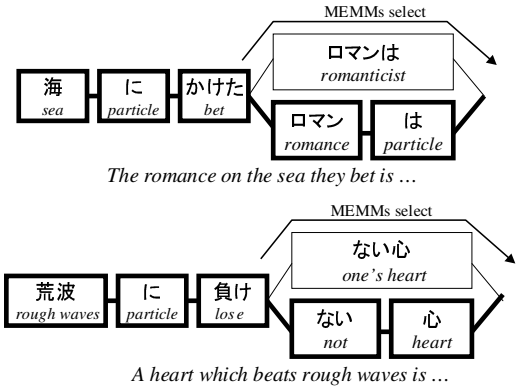

Figure 3: Errors with MEMMs
(Correct paths are marked with bold boxes.)

Table 5: Number of errors in KC dataset

	# of *l*-errors	# of *s*-errors
CRFs	79 (40%)	120 (60%)
HMMs	306 (44%)	387 (56%)
MEMMs	416 (70%)	183 (30%)

l-error: output longer token than correct one
s-error: output shorter token than correct one

this table. *l*-error (or *s*-error) means that a system incorrectly outputs a longer (or shorter) token than the correct token respectively. By length bias, long tokens are preferred to short tokens. Thus, larger number of *l*-errors implies that the result is highly influenced by the length bias.

While the relative rates of *l*-error and *s*-error are almost the same in HMMs and CRFs, the number of *l*-errors with MEMMs amounts to 416, which is 70% of total errors, and is even larger than that of naive HMMs (306). This result supports our claim that MEMMs is not sufficient to be applied to Japanese morphological analysis where the length bias is inevitable.

4.3.2 CRFs and Extended-HMMs

Asahara et al. extended the original HMMs by 1) position-wise grouping of POS tags, 2) word-level statistics, and 3) smoothing of word and POS level statistics (Asahara and Matsumoto, 2000). All of these techniques are designed to capture hierarchical structures of POS tagsets. For instance, in the position-wise grouping, optimal levels of POS hierarchies are changed according to the contexts. Best hierarchies for each context are selected by hand-crafted rules or automatic error-driven procedures.

CRFs can realize such extensions naturally and straightforwardly. In CRFs, position-wise grouping and word-POS smoothing are simply integrated into a design of feature functions. Parameters λ_k for each feature are automatically configured by general maximum likelihood estimation. As shown in Table 2, we can employ a number of templates to capture POS hierarchies. Furthermore, some overlapping features (e.g., forms and types of conjugation) can be used, which was not possible in the extended HMMs.

4.3.3 L1-CRFs and L2-CRFs

L2-CRFs perform slightly better than L1-CRFs, which indicates that most of given features (i.e., overlapping features, POS hierarchies, suffixes/prefixes and character types) are relevant to both of two datasets. The numbers of active (non-zero) features used in L1-CRFs are much smaller (about 1/8 - 1/6) than those in L2-CRFs: (L2-CRFs: 791,798 (KC) / 580,032 (RWCP) v.s., L1-CRFs: 90,163 (KC) / 101,757 (RWCP)). L1-CRFs are worth being examined if there are some practical constraints (e.g., limits of memory, disk or CPU resources).

5 Conclusions and Future Work

In this paper, we present how conditional random fields can be applied to Japanese morphological analysis in which word boundary ambiguity exists. By virtue of CRFs, 1) a number of correlated features for hierarchical tagsets can be incorporated which was not possible in HMMs, and 2) influences of label and length bias are minimized which caused errors in MEMMs. We compare results between CRFs, MEMMs and HMMs in two Japanese annotated corpora, and CRFs outperform the other approaches. Although we discuss Japanese morphological analysis, the proposed approach can be applicable to other non-segmented languages such as Chinese or Thai.

There exist some phenomena which cannot be analyzed only with bi-gram features in Japanese morphological analysis. To improve accuracy, tri-gram or more general n-gram features would be useful. CRFs have capability of handling such features. However, the numbers of features and nodes in the lattice increase exponentially as longer contexts are captured. To deal with longer contexts, we need a practical feature selection which effectively trades between accuracy and efficiency. For this challenge, McCallum proposes an interesting research avenue to explore (McCallum, 2003).

Acknowledgments

We would like to thank Kiyotaka Uchimoto and Masayuki Asahara, who explained the details of their Japanese morphological analyzers.

References

Masayuki Asahara and Yuji Matsumoto. 2000. Extended models and tools for high-performance part-of-speech tagger. In *Proc of COLING*, pages 21–27.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ci You Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(6):1190–1208.

Stanley F. Chen and Ronald. Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University.

Joshua Goodman. 2004. Exponential priors for maximum entropy models. In *Proc. of HLT/NAACL*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(3, (Ser. B)):503–528.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *In Proc. of CoNLL*.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy markov models for information and segmentation. In *Proc. of ICML*, pages 591–598.

Andrew McCallum. 2003. Efficiently inducing features of conditional random fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*.

Fuchun Peng and Andrew McCallum. 2004. Accurate information extraction from research papers. In *Proc. of HLT/NAACL*.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields (to appear). In *Proc. of COLING*.

Simon Perkins, Kevin Lacker, and James Thiler. 2003. Grafting: Fast, incremental feature selection by gradient descent in function space. *JMLR*, 3:1333–1356.

Della Pietra, Stephen, Vincent J. Della Pietra, and John D. Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. 2003. Table extraction using conditional random fields. In *In Proc. of SIGIR*, pages 235–242.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proc. of HLT-NAACL*, pages 213–220.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 2001. The unknown word problem: a morphological analysis of Japanese using maximum entropy aided by a dictionary. In *Proc. of EMNLP*, pages 91–99.

Kiyotaka Uchimoto, Chikashi Nobata, Atsushi Yamada, Satoshi Sekine, and Hitoshi Isahara. 2002. Morphological analysis of the spontaneous speech corpus. In *Proc of COLING*, pages 1298–1302.

Kiyotaka Uchimoto, Chikashi Nobata, Atsushi Yamada, and Hitoshi Isahara Satoshi Sekine. 2003. Morphological analysis of a large spontaneous speech corpus in Japanese. In *Proc. of ACL*, pages 479–488.