Conditional Random Fields を用いた日本語形態素解析

工藤 拓 †[1]　　山本 薫 ‡　　松本 裕治 †

† 奈良先端科学技術大学院大学 情報科学研究科
‡CREST JST, 東京工業大学

†{taku-ku,matsu}@is.naist.jp, ‡kaoru@lr.pi.titech.ac.jp

本稿では, Conditonal Random Fields (CRF) に基づく日本語形態素解析を提案する. CRF を適用したこれまでの研究の多くは, 単語の境界位置が既知の状況を想定していた. しかし, 日本語には明示的な単語境界が無く, 単語境界同定と品詞同定を同時に行うタスクである日本語形態素解析に CRF を直接適用することは困難である. 本稿ではまず, 単語境界が存在する問題に対する CRF の適用方法について述べる. さらに, CRF が既存手法 (HMM, MEMM) の問題点を自然にかつ有効に解決することを実データを用いた実験と共に示す. CRF は, 階層構造を持つ品詞体系や文字種の情報に対して柔軟な素性設計を可能にし, label bias や length bias を低減する効果を持つ. 前者は HMM の欠点であり, 後者は MEMM の欠点である. また, 2 つの正則化手法 (L1-CRF/L2-CRF) を適用し, それぞれの性質について論じる.

Applying Conditional Random Fields to Japanese Morphologiaical Analysis

Taku Kudo †　Kaoru Yamamoto ‡　Yuji Matsumoto †

†Graduate School of Information Science, Nara Institute of Science and Technology
‡CREST JST, Tokyo Institute of Technology

†{taku-ku,matsu}@is.naist.jp, ‡kaoru@lr.pi.titech.ac.jp

This paper presents Japanese morphological analysis based on Conditional Random Fields (CRF). Previous work in CRF assumed that observation sequence (word) boundaries were fixed. However, word boundaries are not clear in Japanese, and hence a straightforward application of CRF is not possible. We show how CRF can be applied to situations where word boundary ambiguity exists. CRF offer an elegant solution to the long-standing problems in Japanese morphological analysis using HMM or MEMM. First, flexible feature designs for hierarchical tagsets become possible. Second, influences of label and length bias are minimized. The former compensate weakness in HMM, while the latter overcomes noticed problems in MEMM. We experiment with CRF, HMM, and MEMM on Japanese annotated corpora, and CRF outperform the other approaches.

1　はじめに

Conditional Random Fields (以下 CRF) [8] は, 系列ラベリング問題のために設計された識別モデル (discriminative model) であり, 正しい系列ラベリングを他の全ラベリング候補と弁別するような学習を行う. 通常の識別モデルとの違いは, 出力が出力集合 $\mathcal{Y}$ の部分集合ではなく, 系列となる点にある. CRF は, 品詞付与 [8], テキストチャンキング [18], 固有表現抽出 [12], HTML からの情報抽出 [15], 書誌データからの情報抽出 [13], といった系列ラベリング問題に適用され, いずれにおいても高い精度を示している.

これまでの CRF の応用の多くは, 出力系列 (e.g., 単語系列) のサイズが固定という状況を想定しており, 各系列のラベル (e.g., 品詞) を正しく判別することに重きが置かれていた. しかし, 日本語や中国語には明示的な単語の境界が存在せず, 出力系列のサイズも未知となる. そのため, 品詞同定と単語境界同定を同時に行うタスクである形態素解析に CRF を直接適用することは困難である. 本稿ではまず, 単語境界が存在する問題への CRF の適用方法について述べる.

CRF は, 従来法である Hidden Markov Models (HMM) (e.g., [4]) や Maximum Entropy Markov Models (MEMM) (e.g., [21]) の問題点を自然にかつ有効に解決する. まず, HMM は系列のための生成モデル (generative model) であり, 正しい系列ラベルを出力する目的で設計されているわけではない. また, 素性の設計に強い制限が加わる. 具体的には, HMM では, 階層構造を持つ品詞体系, オーバラップ

[1]平成 16 年 4 月より NTT コミュニケーション科学基礎研究所リサーチアソシエイト (taku@cslab.kecl.ntt.co.jp)

する素性, 周辺の単語情報, 文字種や文字そのものといった情報を柔軟に取り込むことが困難である．日本語形態素解析に対するこれらの有効性は直感的にも明らかであるが, HMM を用いた解析手法の多くはこれらを無視していた．次に, MEMM は, 識別モデルであり, 多くの情報を用いた学習が可能であるが, label bias [8] や length bias (単語境界の曖昧性から生じる bias) に弱いという欠点を持つ．MEMM は, その解析 (デコード) 時に, 曖昧性の小さい系列を選びやすい (それらに bias がかかる)．これらの問題は, 日本語形態素解析において特に深刻な問題である．その理由として, 1) 品詞体系が複雑で, 曖昧性の分散が大きい (極端に曖昧性の小さい系列が存在する), 2) 単語境界も同定する必要があり, 分割数の小さい (曖昧性の小さい) 系列を選びやすい, などがある．

さらに, CRF のパラメータ推定時に Gaussian prior (L2), Laplacian prior (L1) の 2 つの正則化手法を適用し, それぞれの特性を検証する．Laplacian prior は, 一般的に使われる Gaussian Prior と比べ同程度か若干劣る精度を示す一方で, 不必要な素性を排除し, コンパクトなモデルを構築できる．

以下, CRF を日本語形態素解析に適用する動機付けについて, 従来法と比較しながら 2 章で述べる．次に, CRF の詳細とそのパラメータ推定法について 3 章で紹介する．4,5 章にて, タグ付きコーパスを用いた実験結果, 今後の課題について述べる．

2 日本語形態素解析

2.1 単語境界の曖昧性

明示的な単語境界が無い言語において, それらの同定は言語処理を行ううえで重要なタスクである．最も単純な手法は, 文字単位のチャンキングとして定式化することであるが, 辞書情報を統合しにくく, 曖昧性が大きくなり速度的な問題が残る．

歴史的に日本語形態素解析では, 辞書の存在を仮定することが多い．辞書は, 語彙項目 (単語) とそれに対応する可能な品詞を列挙したデータである．辞書を用いることで, 入力文に対する可能な系列ラベリングを表現した**形態素ラティス**を効率良く構築できる．もし, 辞書にマッチする単語が存在せず, ラティスの構築に失敗した場合は, 別の**未知語処理**が起動される．これは, 一種の仮想的な辞書とみなすことができ, 日本語形態素解析では, 字種情報 (ひらがな, カタカナ, 数値, 漢字など) によるヒューリスティックスを用いて品詞/単語境界候補を動的に生成する．本稿の目的は未知語処理ではないため, 既存のヒューリステックスを用いて候補を生成し, 如何なる入力文に対しても形態素ラティスが構築されるものと仮定する．

図 1 に形態素ラティスの例を示す．この例では, 合計 6 通りの出力系列表現される．太枠で囲んだ系列が正解である．英語等の品詞付与問題と異なり, 単語境界の曖昧性が存在し, 出力系列のサイズが可変になることに注意されたい．

形式的に日本語形態素解析は以下のように定式化される． $\mathbf{x}$ を入力文, $\mathbf{y}$ を 1 つの出力系列, $\mathcal{Y}(\mathbf{x})$ を形態素ラティス中に埋めこまれる系列候補集合とする．$\mathbf{y}$ は, 単語と品詞のペアの系列 $\mathbf{y} = (\langle w_1, t_1\rangle, \ldots, \langle w_{\#\mathbf{y}}, t_{\#\mathbf{y}}\rangle)$ (ただし $\#\mathbf{y}$ は系列のサイズ) として表現される．日本語形態素解析は, 正しい系列 $\hat{\mathbf{y}}$ を全候補集合 $\mathcal{Y}(\mathbf{x})$ から 1 つ選択するタスクとなる．

日本語形態素解析の他の系列ラベリング問題との違いは, 出力系列 $\mathbf{y}$ のサイズが個々の系列によって変化することである．これは中国語やタイ語といった明示的な単語境界がない言語の機械処理における共通の性質である．

2.2 日本語形態素解析の難しさ

2.2.1 階層的品詞体系

日本語形態素解析器として広く認知されているシステムに ChaSen[2] および JUMAN[3] がある．これらは共に階層的な品詞体系を用いている．例えば, ChaSen が用いる IPA 品詞体系[4] では, 品詞は 3 つの属性 (品詞, 活用型, 活用形) から構成される．活用型や活用形は, 動詞, 形容詞といった活用する語のみに与えられる．品詞は, 最大 4 階層の情報を持つ．最上位は名詞, 動詞.. といった大分類である．名詞は一般名詞, 固有名詞.. 等に細分化され, 固有名詞は人名, 地名, 組織名.. 等にさらに細分化される．階層の葉には単語 (基本形) が置かれる．これらの階層を全展開すると, 単語の階層を除いてその数は 500 にも達する．これは Penn Treebank 等の英語の品詞体系に比べると非常に大きい．

これまで, 階層構造を持つ品詞体系, ならびに文字種や単語の部分文字列といった単語そのものの情報をいかにして統合し, 解析精度向上に繋げるかに研究の重点が置かれてきた．最も細い階層を用いるとデータスパースネスの問題を避けられない．逆に最上位の階層を用いると, 品詞の細い違いを区別できない．例えば,「さん」や「君」という名詞性の接尾辞は人名の後に置かれることが多く, 人名の品詞を同定するために有効である．

浅原らは, 1) 連接位置 (前件, 後件) 毎の品詞グルーピング規則, 2) 語彙情報の利用, 3) 語彙と品詞のスムージング, を用い HMM を拡張している [4]．しかし, オーバラップする素性集合, 文字列や文字種といった情報は用いていない．また, スムージング係数も ad-hoc に決定されている．

2.2.2 Label Bias と Length Bias

Maximum Entropy Markov Models (MEMM) [11, 16] や汎用的な識別モデル (e.g., SVMs) を順次適用

[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/

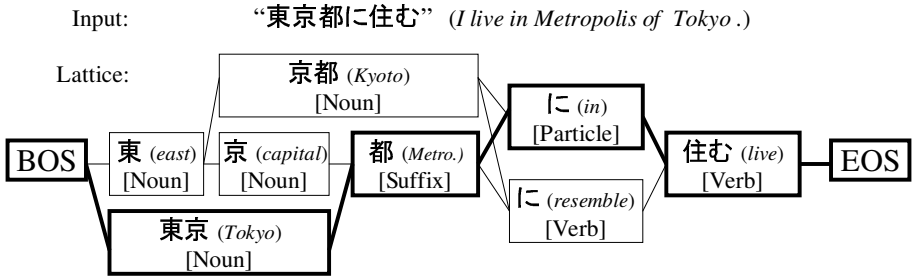

図 1: 形態素ラティス

するモデルは後述する label bias [8] や length bias に弱いとされている.

MEMM では, 系列の各ラベルは, 最大エントロピーモデル (Maximum Entropy Models) の順次適用で決定される. つまり, 元の系列ラベリング問題を個々のラベルを付与するという部分問題に分け, 個々のラベリングの確率の積を全体のラベリングに対する確率と近似する. ただし, 各ラベルは, n–1 個前までのラベルに依存する (n-gram). 形式的に, MEMM は以下の条件付き確率を最大にするような系列を選択する (2-gram の場合). $P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{\#\mathbf{y}} p(\langle w_i, t_i\rangle|\langle w_{i-1}, t_{i-1}\rangle)$.

図 2:(a) に label bias の典型的な例を示す. たとえ MEMM が個々の部分問題にて正しい系列 A-D を推定できたとしても, 全体では B-E の確率が大きくなってしまう. これは, B の後続するラベルは E のみであり, B-E の連接確率が常に 1.0 になってしまうことに起因する. 系列の本質的な曖昧さはラティスを通る「経路」に依存し, 実際には曖昧性の小さい経路が選ばれやすくなる. これが label bias の一般的な性質と言える. MEMM の学習 (エンコード) 時には, 正解の経路のみを考慮し, 個々の部分問題が独立に学習される. そのため, 学習時に考慮されなかった系列の確率値は, 解析 (デコード) 時には非常に不安定になる. 結果として, 経路上の曖昧性の影響を受けやすくなってしまう.

さらに日本語形態素解析には length bias が存在する. これは, サイズの小さい系列 (短い経路) が大きい系列 (長い経路) に比べて選ばれやすくなるという問題である. たとえ個々の連接確率が小さくても, 出力系列サイズが小さい場合, 全体の確率が大きく見積もられてしまう (図 2:(b)). 日本語形態素解析には **最長一致**や**文節数最小**という強いヒューリスティックスが存在するために length bias の存在を無視してもそれなりの解析精度が得られていた. しかし, length bias を無視できない例が存在するのも事実である.

内元らは MEMM を拡張したモデルを日本語形態素解析に適用している [21, 20, 19]. 単語の部分文字列や文字種といった多くの情報を使い, 未知語の精度向上に成功しているが, 既知語の精度は label bias や length bias の影響を少なからず受けている. HMM や ルールベースの手法で解析できる文も MEMM では正しく解析できない場合が報告されている. これらの例については, 4.3.1 章で述べる.

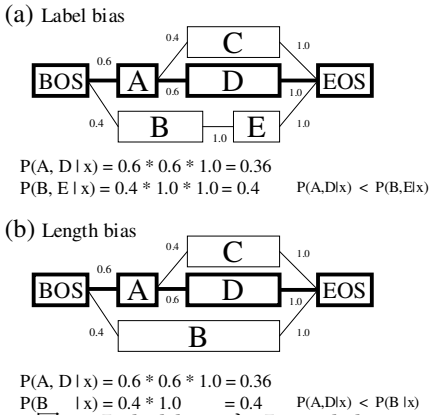

図 2: Label bias と Length bias

3 Conditional Random Fields

Conditional Random Fields (CRF) [8] は, 2.2 節で述べた 2 つの問題を自然にかつ効果的に解消する. CRF は, 識別モデルであるために, 階層構造を持つ品詞体系, 文字種や文字列といった情報を柔軟に取り込むことが可能である. MEMM が元の系列ラベリング問題を部分問題に分割していたのに対し, CRF では, 1 つの指数分布モデル (a.k.a., 最大エントロピーモデル) によって各出力系列 $\mathbf{y}$ の入力文 $\mathbf{x}$ に対する条件付き確率 $P(\mathbf{y}|\mathbf{x})$ を表現する. これにより, 正しい出力系列を他の全出力候補と弁別するような学習が行われる. 学習時に他の全候補を考慮する点が MEMM との大きな違いであり, これこそが label bias や length bias を低減させる効果を生む.

日本語形態素解析では, ある出力系列 $\mathbf{y} = (\langle w_1, t_1\rangle, \ldots, \langle w_{\#\mathbf{y}}, t_{\#\mathbf{y}}\rangle)$ の入力文 $\mathbf{x}$ に対する条件付き確率は以下のように与えられる.

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

ただし $Z_{\mathbf{x}}$ は全系列を考慮したときに確率の和が 1 になるようにするための正規化項である. すなわち,

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

となる. $f_k(\langle w_{i-1}, t_{i-1}\rangle, \langle w_i, t_i\rangle)$ は i 番目と i–1 の出力ラベルに依存する任意の素性関数である. 通常は,

以下のような素性の有無を返す 2 値関数を与えるが, 実数値を返す関数でも構わない[5].

$$f_{1234}(\langle w', t'\rangle, \langle w, t\rangle) \stackrel{\text{def}}{=} \begin{cases} 1 & t' = \text{名詞 \& } t = \text{助詞} \\ 0 & otherwise. \end{cases}$$

$\lambda_k (\in \mathbf{\Lambda} = \{\lambda_1, \ldots, \lambda_K\} \in \Re^K)$ は, 素性関数 f_k に対する重み (パラメータ) である. 我々の定式化では, 位置インデックス i が個々の出力 $\mathbf{y}$ に依存する形となる. 通常の CRF の表記では, 単語境界の曖昧性が存在しないために i は入力 $\mathbf{x}$ が与えられると, どの $\mathbf{y}$ に対しても同一の位置を差し示す.

ここで, 理解を助けるために, 大域素性ベクトル $\mathbf{F}(\mathbf{y}, \mathbf{x}) = \{F_1(\mathbf{y}, \mathbf{x}), \ldots, F_K(\mathbf{y}, \mathbf{x})\}$, を与える. ただし, $F_k(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{\#\mathbf{y}} f_k(\langle w_{i-1}, t_{i-1}\rangle, \langle w_i, t_i\rangle)$ である. 大域素性ベクトルを用いると, $P(\mathbf{y}|\mathbf{x})$ は $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp(\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}))$ と書ける.

入力 $\mathbf{x}$ に対する最適な出力 $\hat{\mathbf{y}}$ は

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} P(\mathbf{y}|\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}) \qquad (1)$$

となる. この最適系列は, Viterbi アルゴリズムを用いて効率良く探索できる. ここで, 便宜的に $\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x})$ を系列 $\mathbf{y}$ のコストと呼ぶこととする.

3.1 パラメータ推定

CRF は, 一般的な最尤推定を用いてパラメータを選択する. つまり, 学習データ $T = \{\langle \mathbf{x}_j, \mathbf{y}_j\rangle\}_{j=1}^N$ に対する対数尤度 $\mathcal{L}_\mathbf{\Lambda}$ の最大化を行う.

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

$\mathcal{L}_\mathbf{\Lambda}$ を大きくするには, 各学習データ $\langle \mathbf{x}_j, \mathbf{y}_j\rangle$ に対し, $\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)} \exp\big(\mathbf{\Lambda} \cdot [\mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \mathbf{F}(\mathbf{y}, \mathbf{x}_j)]\big)$ を大きくすればよい. これは正解のパスのコスト $\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}_j, \mathbf{x}_j)$ と, 残りの全候補のコスト $\mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}_j)$, $\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)$ との「差」をできるだけ大きくすることに相当する. これにより, ラティス中の全候補系列から正解の系列のみを分別するような効果が生まれる.

目的関数の凸性から, 最適点では以下が成立する.

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

ただし, $O_k = \sum_j F_k(\mathbf{y}_j, \mathbf{x}_j)$ は素性 k の学習データ T における出現頻度である. $E_k = \sum_j E_{P(\mathbf{y}|\mathbf{x}_j)}\big[F_k(\mathbf{y}, \mathbf{x}_j)\big]$ は, 素性の k のモデル分布における出現期待値である. この期待値の計算は, 単純には出力系列の候補 $\mathcal{Y}(\mathbf{x})$ を陽に列挙することで実現できるが, その候補数が入力文に対し指数的に増えるために, 事実上困難である. しかし, 動的計画法の一種である forward-backward アルゴリズムと同種の手法で効率良く計算できる.

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

ただし, f_k^* は $f_k(\langle w', t'\rangle, \langle w, t\rangle)$ の短縮形であり, $\mathcal{B}(\mathbf{x})$ は $\mathbf{x}$ の形態素ラティス上に出現するサイズ 2 の全部分系列 (bi-gram) の集合である. $\alpha_{\langle w,t\rangle}$ および $\beta_{\langle w,t\rangle}$ は forward-backward コストであり, 以下の再帰的な定義で与えられる.

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

ただし, $LT(\langle w, t\rangle)$, (もしくは $RT(\langle w, t\rangle)$) は出力 $\langle w, t\rangle$ に左 (もしくは 右) から接続するラベルの集合である. 文頭, 文末を示す 2 つの仮想的なラベルのコスト $\alpha_{\langle w_{bos}, t_{bos}\rangle}$,$\beta_{\langle w_{eos}, t_{eos}\rangle}$ は 1 に初期化しておく. 以上を用いると, 正規化項は $Z_\mathbf{x} = \alpha_{\langle w_{eos}, t_{eos}\rangle} (= \beta_{\langle w_{bos}, t_{bos}\rangle})$ となる.

最尤推定はしばしば過学習の問題を引き起こす. そこで, 過学習を防ぐために, パラメータの正則化を行う. これは事後確率最大化 (MAP) とも呼ばれ, パラメータの事前分布を考慮する最尤推定の一般形である. 事前分布を一様分布にすると, 通常の最尤推定と同一になる. 本稿では, Gaussian Prior (L2-norm) [6], および Laplacian Prior (L1-norm) [7] の 2 つの事前分布を考える [6]. 正則化を行った場合, 目的関数は以下のようになる.

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

以下, L1-norm, L2-norm の正則化を適用した CRF をそれぞれ L1-CRF, L2-CRF と呼ぶ. $C \in \Re^+$ は, CRF のハイパーパラメータであり, モデルの複雑さと学習データに対する適用度をコントロールする. C は, 交差検定等の一般的なモデル選択手法で選択する.

L1-CRF は, 以下の制約付き最適化問題として定式化できる.

$$\max: \quad C \sum_j \log(P(\mathbf{y}_j|\mathbf{x}_j)) - \sum_k (\lambda_k^+ + \lambda_k^-)/2$$

[5] tri-gram や, さらに一般的な n-gram を考慮する素性関数 (e.g., $f_k(\langle w_{i-n+1}, t_{i-n+1}\rangle, \ldots, \langle w_i, t_i\rangle)$) を与えることもできる.

[6] L1-norm による正則化は, 風間らの *Inequality Constraints* の特殊形となる. 本稿では L2-norm との一貫性から L1-norm と呼ぶ.

$$\begin{aligned} & \text{where} \quad \lambda_k = \lambda_k^+ - \lambda_k^- \\ \text{s.t.,} \quad & \lambda_k^+ \geq 0,\ \lambda_k^- \geq 0. \end{aligned}$$

最適点では, 以下の Karush-Kuhun-Tucker 条件が成立する. $\lambda_k^+ \cdot [C \cdot (O_k - E_k) - 1/2] = 0, \lambda_k^- \cdot [C \cdot (E_k - O_k) - 1/2] = 0$, $|C \cdot (O_k - E_k)| \leq 1/2$. これら条件より, $|C \cdot (O_k - E_k)| < 1/2$ の時, λ_k^+ と λ_k^- が共に 0 (つまり, $\lambda_k = 0$) となり, $|C \cdot (O_k - E_k)| = 1/2$ の時のみ, 非 0 の値が λ_k に与えられることが分かる. つまり, C を小さくすれば, $\lambda_k = 0$ となる素性が多くなり, よりコンパクトなモデルを構築できる.

L2-CRF は, $\frac{\delta \mathcal{L}_{\mathbf{\Lambda}}}{\delta \lambda_k} = C \cdot (O_k - E_k) - \lambda_k = 0$ の時に最適点となる. 証明は省略するが, $(O_k - E_k) \neq 0$ が成立し, 全 λ_k に非 0 の値が与えられる.

L1-CRF と L2-CRF の具体的な精度は, 応用や学習データに依存する. L1-CRF の利点は, 不必要な素性に 0 の重みを与え, コンパクトなモデルを構築できる点にある[7] [8]. L1-CRF は, 実応用での制約 (メモリ, ディスク, CPU の制限など) が存在する場合, 有効に機能するであろう.

L2-CRF の最適解は, 古典的な iterative scaling algorithms (e.g., IIS や GIS [14]) や, 準ニュートン法 (e.g., L-BFGS [9]) を用いて導出できる. L1-CRF に対しては, 制約付き最適化手法 (e.g., L-BFGS-B [5]) が適用できる.

3.2 CRF と系列のための線形識別モデル

解析 (デコード) 時に用いられる式 (1) は出力系列から導出される素性ベクトル $\mathbf{F}(\mathbf{y}, \mathbf{x})$ とパラメータ $\mathbf{\Lambda}$ の内積, すなわち一般的な線形モデルとして表現されていることが分かる.

HMM は, 単語生起確率の対数 $\log p(w|t)$ と連接確率の対数 $\log p(t|t')$ をパラメータ $\lambda_{\langle w,t \rangle}, \lambda_{\langle t,t' \rangle}$ とみなせば, 線形モデルとして定式化できる.

$$\begin{aligned} & \log(\prod_{i=1}^{\#\mathbf{y}} p(w_i|t_i) p(t_i|t_{i-1})) \\ =& \sum_{i=1}^{\#\mathbf{y}} [\log p(w_i|t_i) + \log p(t_i|t_{i-1})] \\ =& \sum_{i=1}^{\#\mathbf{y}} \Big[\sum_{\langle w,t \rangle} I(w = w_i, t = t_i) \log p(w|t) \\ & \quad + \sum_{\langle t,t' \rangle} I(t = t_i, t' = t_{i-1}) \log p(t|t') \Big] \\ =& \sum_{\langle w,t \rangle} F_{\langle w,t \rangle}(\mathbf{y}, \mathbf{x}) \cdot \lambda_{\langle w,t \rangle} + \sum_{\langle t,t' \rangle} F_{\langle t,t' \rangle}(\mathbf{y}, \mathbf{x}) \cdot \lambda_{\langle t,t' \rangle} \\ =& \mathbf{\Lambda} \cdot \mathbf{F}(\mathbf{y}, \mathbf{x}) \end{aligned}$$

ただし, $I(\cdot)$ は indicator function, $F_{\langle w,t \rangle}(\mathbf{y}, \mathbf{x}), F_{\langle t,t' \rangle}(\mathbf{y}, \mathbf{x})$ は大域素性ベクトルの 1 つの次元に相当し,

$$\begin{aligned} F_{\langle w,t \rangle}(\mathbf{y}, \mathbf{x}) &= \sum_i I(w = w_i, t = t_i), \\ F_{\langle t,t' \rangle}(\mathbf{y}, \mathbf{x}) &= \sum_i I(t = t_i, t' = t_{i-1}) \end{aligned}$$

となる. すなわち, HMM は, 素性関数が単語生起と連接に限られる線形モデルと解釈できる. 上記の議論は, 広く認知される**コスト最大 (最小) 法**にもあてはまる. 人手で与えられたコスト値が, 生成モデルである HMM により統計的に算出できるようなった歴史的背景がある. CRF もコスト最大 (最小) 法を踏襲し, 統計的にコストを算出するが, 識別モデルを基礎としていることと, 単語生起/連接コストの 2 つの観点を, 部分文字列, 文字種, 周辺の単語といった別の観点にまで拡張できる点が異なる.

さらに, ロス関数の定義から多種多様な線形モデルが導出できるのと同様な議論が系列ラベリング問題にもあてはまる. つまり, 最大エントロピーモデルが CRF に対応するのと同様, SVM や Boosting のロス関数を用いた系列ラベリングモデルを与えることができる. CRF, SVM, Boosting の (正則化項を含まない) 経験的ロス関数 $\mathcal{E}_{CRF}, \mathcal{E}_{SVM}, \mathcal{E}_{Bo}$ は以下のようになる.

$$\begin{aligned} \mathcal{E}_{CRF} &= -\sum_j \Big[\log \Big(\sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)} \exp \left(\mathbf{\Lambda} \cdot [\mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \mathbf{F}(\mathbf{y}, \mathbf{x}_j)] \right) \Big) \Big] \\ \mathcal{E}_{SVM} &= -\sum_j \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)} \max(0, 1 - \mathbf{\Lambda} \cdot [\mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \mathbf{F}(\mathbf{y}, \mathbf{x}_j)]) \\ \mathcal{E}_{Bo} &= -\sum_j \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)} \exp \left(\mathbf{\Lambda} \cdot [\mathbf{F}(\mathbf{y}_j, \mathbf{x}_j) - \mathbf{F}(\mathbf{y}, \mathbf{x}_j)] \right) \end{aligned}$$

Altun らは, それぞれについてパラメータの推定方法を導出し, SVM のロス関数を用いる HM-SVM (Hidden Markov SVM)[3] が精度的に優れていると報告している [2, 1].

パラメータ推定時における各モデルの違いは, 全候補集合 $\mathbf{y} \in \mathcal{Y}(\mathbf{x}_j)$ の扱い方に表われる. さきに述べたように, CRF は動的計画法を用いて効率良く列挙可能であったが, SVM や Boosting のロス関数は, このような手法が使えず, 学習のコストが大きくなる問題がある. CRF は, 精度と規模耐性とのバランスがとれた手法と言える.

4 実験と考察

4.1 実験設定

CRF の有効性を示すために, 京都大学テキストコーパス ver. 2.0 (**KC**) と RWCP テキストコーパス (**RWCP**) の 2 つのタグ付きコーパスを用いて実験を行った. これらの 2 つのコーパスは異なる品詞体系でタグ付けされている. データの詳細を表 1 にまとめる.

CRF の 1 つの利点として, オーバラップする素性や文字種や部分文字列といった素性を素性関数とい

[7] 一般論として, L1-norm は, 素性集合中に有効な素性が少ないとき, L2-norm は多いときに良い精度を示す傾向がある.

[8] L1-norm と L2-norm の各正則化手法は, Boosting と Support Vector Machines に関連があることが指摘されている [17].

表 1: 実験データの詳細

	KC	**RWCP**
ソース	毎日新聞 ('95)	毎日新聞 ('94)
辞書 (活用等を全展開した語彙数)	JUMAN ver. 3.61 (1,983,173)	IPADIC ver. 2.7.0 (379,010)
品詞体系のサイズ	2 階層の品詞, 活用型, 活用形, 基本形	4 階層の品詞, 活用型, 活用形, 基本形
文数 (学習)	7,958 (1 月 1 日 - 1 月 8 日の記事)	10,000 (先頭の 1 万文)
形態素数 (学習)	198,514	265,631
文数 (テスト)	1,246 (1 月 9 日の記事)	25,743 (残りすべて)
形態素数 (テスト)	31,302	655,710
素性数	791,798	580,032

う形で柔軟に投入できることにある．このような柔軟な素性設計は HMM では困難である．表 2 にデータセット **KC** にて使用した素性関数のテンプレートをまとめる．例えば, テンプレート $\langle bw, p1\rangle$ からは, (語彙 × 品詞) 個の素性関数が生成され, 各関数は以下のような 2 値を返す.

$$f_{1234}(\langle w', t'\rangle, \langle w, t\rangle) \stackrel{\text{def}}{=} \begin{cases} 1 & bw = \text{は} \ \& \ p1 = \text{助詞} \\ 0 & otherwise. \end{cases}$$

RWCP のテンプレートは, 品詞の階層のサイズが異なることを除けば **KC** のそれと本質的に同一である．もし着目する語が**語彙化**されている場合, つまり語の品詞が助詞, 助動詞, 接尾辞,「する, 言う」等の頻出動詞の場合は, 語彙レベルのテンプレートも用いる．このような語彙化は, 日本語形態素解析において頻繁に用いられる．未知語処理によって生成された候補については, 語の長さ, 長さ 2 までの, 接頭/接尾辞, ひらがな/漢字/アルファベットといった文字種を用いる．また, 頻度による足切りなどは行わず, ラティス上に観察されたすべての素性を用いる．各データでの素性数を表 1 の最下行に示す.

評価は, 以下で与えられる F 値 ($F_{\beta=1}$) で行う.

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

さらに, 正解の基準として, 以下の 3 つを設けた. 1) **seg**: 単語区切りのみ正解, 2) **top**: 単語区切りと品詞の大分類が正解, 3) **all**: 全情報が正解.

L1-CRF, L2-CRF のハイパーラメータである C は, 交差検定により設定した．システムは C++ にて実装し, 全実験は XEON 2.8Ghz, 主記憶 4.0Gbyte の Linux で行った．L-BFGS-B および L-BFGS の各最適化アルゴリズムを L1-CRF, L2-CRF のパラメータ推定に用いた.

4.2 結果

表 3, 4 に **KC**, および **RWCP** の実験結果を示す．3 つのレベルの F 値 (*seg/top/all*) を L1-CRF, L2-CRF と同コーパスで実験を行なった bi-gram HMM (ベースライン) についてそれぞれ提示している.

表 2: 素性テンプレート: $f_k(\langle w', t'\rangle, \langle w, t\rangle)$

$t' = \langle p1', p2', cf', ct, bw'\rangle$, $t = \langle p1, p2, cf, ct, bw\rangle$, ただし $p1'/p1$, $p2'/p2$, cf'/cf, ct'/ct bw'/bw は, それぞれ 語 w'/w の品詞大分類, 再分類, 活用型, 活用形, 基本形である

タイプ	テンプレート
uni-gram	$\langle p1\rangle$
基本素性	$\langle p1, p2\rangle$
w 既知	$\langle bw\rangle$
	$\langle bw, p1\rangle$
	$\langle bw, p1, p2\rangle$
w 未知	w の文字列長
	サイズ 2 までの接尾 × $\{\phi, \langle p1\rangle, \langle p1, p2\rangle\}$
	サイズ 2 までの接頭 × $\{\phi, \langle p1\rangle, \langle p1, p2\rangle\}$
	文字種 × $\{\phi, \langle p1\rangle, \langle p1, p2\rangle\}$
bi-gram	$\langle p1', p1\rangle$
基本素性	$\langle p1', p1, p2\rangle$
	$\langle p1', p2', p1\rangle$
	$\langle p1', p2', p1, p2\rangle$
	$\langle p1', p2', cf', p1, p2\rangle$
	$\langle p1', p2', ct', p1, p2\rangle$
	$\langle p1', p2', cf', ct', p1, p2\rangle$
	$\langle p1', p2', p1, p2, cf\rangle$
	$\langle p1', p2', p1, p2, ct\rangle$
	$\langle p1', p2', p1, p2, cf, ct\rangle$
	$\langle p1', p2', cf', p1, p2, cf\rangle$
	$\langle p1', p2', ct, p1, p2, ct\rangle$
	$\langle p1', p2', cf', p1, p2, ct\rangle$
	$\langle p1', p2', ct', p1, p2, cf\rangle$
	$\langle p1', p2', cf', ct', p1, p2, cf, ct\rangle$
w' 語彙化	$\langle p1', p2', cf', ct', bw', p1, p2\rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, cf\rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, ct\rangle$
	$\langle p1', p2', cf', ct', bw', p1, p2, cf, ct\rangle$
w 語彙化	$\langle p1', p2', p1, p2, cf, ct, bw\rangle$
	$\langle p1', p2', cf', p1, p2, cf, ct, bw\rangle$
	$\langle p1', p2', ct', p1, p2, cf, ct, bw\rangle$
	$\langle p1', p2', cf', ct', p1, p2, cf, ct, bw\rangle$
w'/w 共に語彙化	$\langle p1', p2', cf', ct', bw', p1, p2, cf, ct, bw\rangle$

KC データセットについては, 内元らの MEMM [21] の結果, ルールベースのシステム JUMAN[9]の結果も載せている． 公平な評価にするために, CRF, MEMM, HMM-bigram は同じコーパスを用いて実験を行っている.

RWCP データセットについては, 浅原らの Extended HMM (E-HMM) の結果も示す. E-HMM も CRF と同じコーパスを用いて実験を行った. E-HMM は, 現在の ChaSen に用いられている手法である.

[9]JUMAN は,「未知語」という品詞を辞書に記載されていない単語に付与する．このような語は「名詞-サ変」というデフォルト品詞を与えて評価した.

表 3: 実験結果: KC

system	$F_{\beta=1}$ (*seg* / *top* / *all*))
L1-CRF ($C=3.0$)	98.80 / 98.14 / 96.55
L2-CRF ($C=1.2$)	98.96 / 98.31 / 96.75
HMM-bigram	96.22 / 94.99 / 91.85
MEMM (Uchimoto 01)	96.44 / 95.81 / 94.27
JUMAN (rule-based)	98.70 / 98.09 / 94.35

表 4: 実験結果: RWCP

system	$F_{\beta=1}$ (*seg* / *top* / *all*))
L1-CRF ($C=3.0$)	99.00 / 98.58 / 97.30
L2-CRF ($C=2.4$)	99.11 / 98.72 / 97.65
HMM-bigram	98.82 / 98.10 / 95.90
E-HMM (Asahara 00)	98.86 / 98.38 / 97.00

結果より, CRF は精度という点で既存手法より優れていることが分かる.

4.3 考察

4.3.1 CRF と MEMM

内元らは MEMM を拡張したモデルを日本語形態素解析に適用している [21, 20, 19]. MEMM は識別モデルであるため, HMM では適用困難であった素性 (文字種や部分文字列) を用いることで未知語の精度を大幅に向上させることに成功している. 一方で, HMM やルールベースのシステム (JUMAN) で正しく解析できる既知語に対して解析に失敗する例が報告されている.

図 3 に内元らの MEMM で誤って解析された例を示す. 正しい解析結果は, 太枠で示される. 内元らは, これらの解析誤りの主たる要因を辞書中の非標準の表記と結論付けている. すなわち, 図 3 では, 「ロマンは」は「ロマン派」,「ない心」は「内心」とそれぞれ表記されるのが一般的であり, これらの辞書の不整合が誤りの原因という主張である.

もちろんこのような不整合は存在しない方がよく, これらが解析誤りの要因の 1 つであることは否定できないが, 図 3 の例は典型的な length bias の影響と考えるのが自然であろう. このような結論に至る背景として, 同一辞書を用いた CRF, HMM, ルールベースのシステム (JUMAN) はこれら 2 つの文を正しく解析できることが挙げられる. length bias により, 短い系列が長い系列より選ばれやすくなる. この例の場合, 「ロマンは」「ない心」はサイズ 1 に対し,「ロマン/は」「ない/心」はサイズ 2 であり, 経路上の曖昧性が小さい前者が選ばれやすい. さらに,「ロマン」と「ロマンは (ロマン派)」は同一の品詞「名詞」を持つ. 結果として, MEMM の場合,「かけた」から「ロマン」と「ロマンは」のそれぞれに繋がる連接確率はほぼ同一になってしまい, 全体の確率値を大きくするためにはサイズの小さい系列 (短い経路) を選ばざるを得ない.

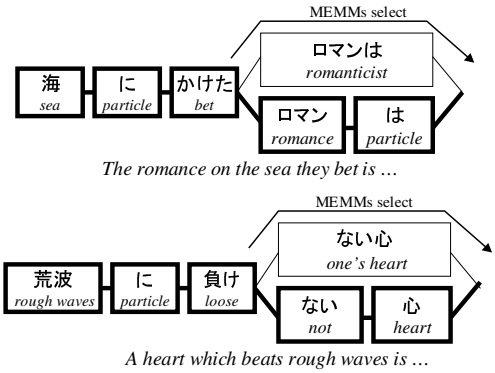

図 3: MEMM による誤り例 (正解は太枠)

4.3.2 CRF と Extended-HMM

浅原らは, 1) 連接位置 (前件, 後件) 毎の品詞グルーピング規則, 2) 語彙情報の利用, 3) 語彙と品詞のスムージング, を用い HMM を拡張している [4]. 精度向上のためには, 階層的な品詞構造から文脈毎に最適な階層を選択したり, 複数の階層をスムージングする必要があり, これらが彼等の拡張の動機付けになっている. 連接位置毎のグルーピングでは, 各文脈毎に, 最適な階層レベルが定義される. これらの定義は, 人手や誤り主導モデルによって半自動的に与えられる.

CRF は, このような拡張を自然にかつ直接的に実現可能である. 連接位置毎のグルーピング, 語彙-品詞のスムージングといった拡張は素性関数の設計という単純な手続きに還元される. CRF のパラメータ λ_k は, 最尤推定により自動的に設定される. 表 2 に示すように, 本実験では, 品詞の階層や語彙の情報を柔軟に取りこむ目的で, 多くの素性関数を用いている. さらに, いくつかの素性関数はオーバラップしており (例えば, 品詞-活用型と品詞-活用形のテンプレートはオーバラップする), このような素性は HMM では適用困難である[10].

4.3.3 L1-CRF と L2-CRF

従来研究の多くは, Gaussian Prior に基づく L2-CRF を用いて過学習を低減させていた. ここでは, L2-CRF と L1-CRF の性質を実験結果を交えながら考察する. 解析精度は L2-CRF の方が若干高かった. しかし, 実際に使われた素性 ($\lambda_k \neq 0$ となる素性) の数は, L1-CRF のほうが約 1/8 - 1/6 程度小さい. (L2-CRF: 791,798 (KC) / 580,032 (RWCP) v.s., L1-CRF: 90,163 (KC) / 101,757 (RWCP)). 3.1 章で示したように, L1-CRF は, 疎なモデルを出力でき, これによりコンパクトな解析システムを構築できる[11].

[10] 包含関係にある素性であれば, 確率値の線型補間によりそれぞれを統合できる.

[11] KC データにて, 実際のモデルのサイズは, L1-norm 7MB, L2-norm 47MB であった.

さらに, L1-CRF は, 決定的に素性を選択するために, どのような素性が有効か, 実際に適用される素性は何かといった分析が行いやすい.

5 おわりに

本稿では, Conditonal Random Fields (CRF) に基づく日本語形態素解析を提案し, 従来手法 (HMM, MEMM) に対する CRF の優位性を示した.

- HMM はそのモデルの制約から, 数多くの素性を柔軟に取り入れることが難しかったが, CRF は可能となる.
- MEMM で問題となる label, length bais に強い.

2 つのタグ付きコーパスを用いた実験により, 上記の利点を検証するとともに, HMM, MEMM に対する精度的優位性を確認した. 本稿では, 日本語の形態素解析に特化していたが, 本提案手法は他の単語境界が存在しない言語, 例えば中国語やタイ語へも適用可能である.

本稿では, bi-gram の素性のみを用いた. しかし, 実際には bi-gram では解析できない事例が存在する. 精度向上には tri-gram や, より一般的な n-gram の素性を投入する必要がある. しかし, これらはラティスの複雑さを指数的に増加させる. そのため, 解析速度という面から考えると, 全 tri-gram を投入することは非現実的である. 一定の解析速度保ちつつ, 長文脈を考慮するには, 精度と速度のバランスをうまく調節できる素性選択手法が必要であろう. L1-CRF は, 決定的な素性選択が可能であり, 精度向上に繋がる有効な素性 (長文脈) を選択するのに適用可能だと考えられる. また, McCallum らは, L2-CRF の枠組みでこのような現実的な素性選択手法を提案している [10]. 今後は, これらの素性選択手法を実際に長文脈の選択に適用したい.

謝辞

MEMM および E-HMM の各形態素解析システムの評価にあたり, CRL の内元清貴氏, NAIST の浅原正幸氏に協力を頂いた. ここに感謝の意を表する.

参考文献

[1] Yasemin Altun and Thomas Hofmann. Large margin methods for label sequence learning. In *Proc. of EuroSpeech*, 2003.

[2] Yasemin Altun, Mark Johnson, and Thomas Hofmann. Investigating loss functions and optimization methods for discriminative learning o f label sequences. In *Proc. of EMNLP*, pages 145–152, 2003.

[3] Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. Hidden markov support vector machines. In *Proc. of ICML*, pages 3–10, 2003.

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

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

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

[7] Jun'ichi Kazama and Jun'ichi Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proc. of EMNLP*, pages 137–144, 2003.

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

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

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

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

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

[13] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proc. of HLT/NAACL*, 2004.

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

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

[16] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*, pages 133–142, 1996.

[17] Gunnar. Rätsch. *Robust Boosting via Convex Optimization*. PhD thesis, Department of Computer Science, University of Potsdam, 2001.

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

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

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

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