# 学習誤り最小化に基づく条件付き確率場の学習: 言語解析への適用

鈴木 潤 † 磯崎 秀樹 †
† 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
〒 619-0237 京都府相楽郡精華町光台 2-4
{jun,isozaki}@cslab.kecl.ntt.co.jp

**概要**

本稿では、誤り最小化 (MCE) に基づく学習の枠組で，条件付き確率場を学習する方法を提案する．この枠組により，誤りを推定するあらゆる評価基準を，学習時の目的関数として利用可能となる．つまり，対象タスクの評価基準を学習時の目的関数として利用可能となり，どのような評価基準のタスクでも，タスクの性能を最大化する学習が可能となる．本稿では，系列セグメンテーションタスク (チャンキング，固有表現抽出) を具体例として，タスク評価基準であるセグメンテーション F 値を条件付き確率場の目的関数として学習する方法を述べ，実験によりその有効性を検証する．

## 1 はじめに

形態素解析，チャンキング，固有表現抽出，係り受け解析といった自然言語解析の問題は，周囲の文脈に依存した構造を学習する問題として定式化するのが自然である．このような周囲の文脈に依存した構造を学習・推定するモデルとして，近年，**条件付き確率場 (Conditional Random Fields: CRFs)** が注目され，これら自然言語解析の問題で良好な結果を示している [1, 2, 3].

CRF は，入力に対する出力の条件付き確率の**尤度最大化 (Maximum Likelihood: ML)** かパラメタの**事後確率最大化 (Maximum A Posterior: MAP)** に基づいて学習をおこうのが一般的である．しかし，例えば，チャンキングや固有表現抽出では，抽出された各セグメント (チャンク) の F 値によって最終的にタスクの性能が評価されるが，ML/MAP 学習では，直接このタスク評価基準を最大化しているわけではない．このように，機械学習手法を実タスクへ適用する際，学習時の目的関数と対象タスクの評価基準で不整合が生じている場合がある．

そこで，本稿では，**(識別) 誤り最小化 (Minimum Classification Error: MCE)** に基づく CRF の学習方法の枠組を提案する．MCE の理論的な枠組により，誤りを推定するあらゆる評価基準を学習時の目的関数 (損失関数) として定式化することが可能となる．つまり，対象タスクの評価基準を学習時の目的関数として定義することが可能である．これにより，学習と評価の不整合を解消でき，素性やモデルを変更しなくても性能の向上が期待できる．具体例として，ここではテキストの系列セグメンテーションタスク (チャンキング，固有表現抽出) に焦点をあて，このタスクで最も一般的なセグメンテーション F 値を目的関数として定式化する方法を述べる．また，実験により有効性を検証する．

## 2 系列セグメンテーションタスク

本稿で提案する手法は，本来，任意の CRF に対する学習手法である．しかし，問題を具体的に議論するために，本稿では**系列セグメンテーション (sequential segmentation)** タスクに絞って議論を行う．系列セグメンテーションタスクとは，品詞タグ付けのように系列データに対してタグを付与する**系列ラベリング (sequential labeling)** タスクの一種で，系列のセグメントを抽出する問題のクラスであると定義する．図 1 に系列セグメンテーションタスクの例として，英語チャンキングと英語固有表現抽出タスクを示す.

Phrase Chunking

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Seg.: | NP | VP | NP | | | | VP | | PP | NP | | | | |
| $x$: | He | reckons | the | current | account | deficit | will | narrow | to | only | # | 1.8 | billion | . |
| $y$: | B-NP | B-VP | B-NP | I-NP | I-NP | I-NP | B-VP | I-VP | B-PP | B-NP | I-NP | I-NP | I-NP | O |

Named Entity Recognition

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Seg.: | ORG | | | PER | | | LOC | |
| $x$: | United | Nation | official | Ekeus | Smith | heads | Baghdad | . |
| $y$: | B-ORG | I-ORG | O | B-PER | I-PER | O | B-LOC | O |

図 1: 系列セグメンテーションタスクの例

系列セグメンテーションタスクでは，IOB タグ[*1]を利用する方法が現在最も一般的かつ簡単な解法である．簡略した説明として，各セグメントの開始位置 (単語) に B(begin)，B に続くセグメント内に I(inside)，対象セグメント以外の位置には O(outside) タグを追加し，複数位置で構成されるセグメントを単純なタギング問題へと置き換える方法である．図 1 中の「Seg.」の行が実際のタスクでのセグメントであり，$y$ が IOB 付きのタグに変換した例で出力系列となる．また，$x$ は $y$ に対応する入力系列 (単語列) である．

系列セグメンテーションタスクの評価には，セグメントの F 値 ($\gamma = 1$)[5] が最も広く使用されている．

$$\begin{aligned} \text{F-score:} &= \frac{(\gamma^2+1)\cdot \text{precision}\cdot \text{recall}}{\gamma^2\cdot \text{precision} + \text{recall}} \\ &= \frac{(\gamma^2+1)\cdot TP}{\gamma^2\cdot FN + FP + (\gamma^2+1)\cdot TP} \end{aligned} \quad (1)$$

$TP, FP, FN$ は，それぞれ true positive，false positive，false negative のセグメント数を表す．このとき IOB の O タグが付与されるセグメントは，抽出対象のセグメントにはカウントされない．つまり，再現率 (recall) の分母は O 以外の対象とするセグメントの総数になる．

次に，CRF の文脈で系列セグメンテーションタスクを考える．CRF で系列セグメンテーションタスクをモデル化する場合，入力変数 $x$ と出力変数 $y$ の要素数が同じ linear

[*1] ここでは IOB2[4] を用いた

chain CRF を用いるのが最も一般的かつ簡単な方法である．CRF の詳細な説明は，現在では多数の文献があるのでそちらに譲る．本稿では，文献 [2] の定義を参照して説明する．入力 $x$ が与えられたときの出力 $y$ の条件付き確率は，パラメタベクトル $\lambda$ を用いて以下のように書ける．

$$p(y|x;\lambda) = \frac{1}{Z_\lambda(x)} \exp(\lambda \cdot F(y,x)) \tag{2}$$

ただし，$F(y,x)$ は CRF の大域的素性，$Z_\lambda(x) = \sum_{y\in\mathcal{Y}} \exp(\lambda \cdot F(y,x))$ は正規化項である．また，CRF の決定関数，つまり $x$ が与えられたときの最尤出力 $\hat{y}$ は，以下の式で与えられる．

$$\hat{y} = \arg\max_{y\in\mathcal{Y}} \lambda \cdot F(y,x) \tag{3}$$

これは，$p(y|x;\lambda) \propto \lambda \cdot F(y,x)$ から導出され，正規化項 $Z_\lambda(x)$ は出力 $y$ に依存しない値なので省略されている．CRF では，条件付き確率の (対数) 尤度最大化 (ML) に基づく学習，つまり式 (2) から導出される対数損失の最小化 ($\arg\min_\lambda -\sum_k \log p(y^{*k}|x^k;\lambda)$) が最も基本的な学習方法であり [1]，実用的には，過学習を防ぐ効果を持つ，パラメタの事後確率最大化 (MAP) に基づく学習 ($\arg\min_\lambda -\sum_k p(\lambda|y^{*k},x^k) \propto \sum_k p(y^{*k}|x^k;\lambda)p(\lambda)$)[2] が用いられる．

このように，ML/MAP 学習の目的関数は，大雑把に言うと系列内の個々の誤り易さの総和を対数損失で平滑化した形となっている．つまり，式 (1) で示した系列セグメンテーションタスクの評価基準との直接的な関係性はない．端的な例としては，B-X, I-X というセグメント X の部分を O, O と間違える場合と B-X, B-X と間違える場合では，前者は TP の数が-1 されるだけであるが，後者は TP-1 と FP+2 となり評価は低くなる．しかし，系列の対数損失の場合には後者の方がタグの間違いは一つであることからも，誤り小さいと判定される可能性を持っている．このように，系列セグメンテーションタスクにおいては，ML/MAP 学習では，タスクの性能を最適化しているとは必ずしも言えない可能性がある．

## 3 損失関数の設計：理論的背景

古典的なパターン識別法であるベイズ決定則と理論的に深い関係を持つ学習法に **(識別) 誤り最小化 (Minimum Classification Error: MCE)** 基準に基づく学習の枠組がある [6, 7]．直感的な説明として，MCE は，決定的学習の一つで経験的誤りを直接最小化する学習の枠組である．つまり，決定関数から導出される経験的誤りを表す (近似) 損失関数を，学習時の目的関数に用いることを提案している．

入力を $x \in \mathcal{X}$，出力を $y \in \mathcal{Y}$ とすると，ベイズ決定則の決定関数は，単純に $\hat{y} = \arg\max_{y\in\mathcal{Y}} g(y,x,\lambda)$ となる．このとき，誤り推定関数は，最も単純には以下の式で計算できる．

$$d(y,x,\lambda) = -g(y^*,x,\lambda) + \max_{y\in\mathcal{Y}\setminus y^*} g(y,x,\lambda). \tag{4}$$

つまり，正解 $y^*$ と正解以外の最尤候補 $y$ の差により誤りを推定することを意味する．ここで，$N$ サンプルの学習データ $T = \{(x^k,y^k)\}_{k=1}^N$ があるとすると，学習データでの経験的損失を最小化する学習は以下の式で定義できる．

$$\arg\min_\lambda \sum_k \delta(d(y^k,x^k,\lambda)) \tag{5}$$

ここで，$\delta(r)$ は $r<0$ のとき 0 を返し，それ以外では 1 を返すステップ関数である．つまりこれは，0-1 損失を最小にするという，学習の基本的な考えを表している．

式 (4) は，一般的にパラメタ $\lambda$ に対して非連続関数になるので，扱い易いように，右辺第二項の max を soft-max ($\max_k r_k \approx \log\sum_k \exp(r_k)$) に置換する，

$$d(y,x,\lambda) = -g(y^*,x,\lambda) + \log\left[\mathcal{A}\sum_{y\in\mathcal{Y}\setminus y^*} \exp(g(y,x,\lambda)\cdot\psi)\right]^{\frac{1}{\psi}} \tag{6}$$

ただし，$\mathcal{A} = \frac{1}{|\mathcal{Y}|-1}$，$\psi$ は $L_\psi$-ノルムを表し正の実数をとる．この関数 $d$ は，$\psi\to\infty$ のとき式 (4) に収束する．注意点として，式 (6) は，式 (4) の近似関数の一例であり，対象タスクに合わせて近似関数を構築すれば良い．

同様に，式 (5) のステップ関数 $\delta()$ で表される 0-1 損失関数も，非連続・微分不可能な関数なので，平滑化関数 $l()$ で置き換える．例えば，0-1 損失には，sigmoid 関数 $\left(\frac{1}{1+\exp(-a\cdot d(y,x,\lambda)-b)}\right)$ を用いて近似する ($a\to\infty$)．ここで，$a$ と $b$ はハイパーパラメタである．$l()$ は，望む目的関数の形に適合するように関数 $d()$ を平滑化する関数である．候補としては sigmoid 以外にも，logistic 回帰や CRF で用いられる対数損失や，boosting の指数損失，更には，SVM の Hinge 損失等も考えられる．これらの平滑化関数は全て 0-1 損失の近似や上限を与える関数として定義されている．

ML/MAP 学習と同様に系列全体に対する損失の平均を目的関数とする場合，MCE 学習の枠組では以下のように書き表すことができる．

$$\mathcal{L}_\lambda^{\mathrm{MCE}} = \frac{1}{N}\sum_{k=1}^N l(d(y^k,x^k,\lambda)) + \frac{||\lambda||^\phi}{\phi C} \tag{7}$$

$N$ は定数なので，消去することも可能である．また，MAP 学習と同様に，過学習を防ぐために $L_\phi$-ノルムを用いた正則化問題として定式化している．

このように MCE 学習の枠組は，理論的には古典的なベイズ決定理論に基づいて構成されており，誤り推定関数を損失関数として最小化することを示している．ここで注目すべき点は，MCE 学習の枠組下で学習時の目的関数を考えると，誤り推定関数 $d()$ と，$d()$ の平滑化関数 $l()$ を様々に選択し組み合わせることで，(実際に学習の目的関数として妥当性があるかは別問題としても) あらゆる形の目的関数を記述することが可能であるという点である．本稿では，この理論的背景を基に，CRF で系列セグメンテーションタスクを学習するのに適した目的関数を設計する．

## 4 セグメンテーション F 値損失関数

系列セグメンテーションタスクの評価基準，式 (1) を学習時の損失関数として定式化する方法を述べる．前節で述べたように，式 (1) を近似するセグメントの誤り推定関数とその平滑化関数を定義する．

まずはじめに，セグメント単位の損失関数を導入する．$s^y = \{s_1^y,\cdots,s_m^y\}$ を $y$ に基づくセグメント系列とする．$\mathcal{S}_i^y$ を，セグメント $s_i^y$ 内の可能な出力変数の系列の集合とする．また，$s_i^y(y)$ をセグメント $s_i^y$ に対応する位置の $y$ の部分出力系列とし，$\mathcal{Y}(s_i^y(y))$ を $s_i^y(y)$ に対応する可能な出力の集合とする．このとき，セグメント単位の損失関数を

以下のように定義する．

$$\mathcal{L}_\lambda^{\text{MCE-S}} = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{|s^{y^{*k}}|} l(d(s_i^{y^{*k}}, x^k, \lambda)) \quad (8)$$

次に，セグメント単位の誤り推定関数を式 (6) に基づいて以下のように定義する．

$$d(s_i^{y^*}, x, \lambda) = -g^* + \frac{1}{\psi}\left[\log\left(Z_\lambda(x,\psi) - \exp(g^*\cdot\psi)\right) + \log\mathcal{A}\right]$$

$$Z_\lambda(x,\psi) = \sum_{y\in\mathcal{Y}}\exp(g\cdot\psi)$$

ただし，$g^* = g(s_i^{y^*}(y^*), x, \lambda)$，$g = g(s_i^{y^*}(y), x, \lambda)$ とする．このときの系列に対する各セグメントの決定関数を以下のように定義する．

$$g(s_i^{y^*}(y), x, \lambda) = \max_{\hat{y}\in\mathcal{Y}(s_i^{y^*}(y))} \lambda\cdot F(\hat{y}, x)$$

未知データに対しては正解セグメントは不明なので，従来の CRF の決定関数を用いた場合でも正解を最適パスと選択するように，大域的な情報を考慮して各セグメントが選択されるような決定関数となっている．

セグメント単位の損失関数を用いると，セグメンテーション F 値損失関数は以下のようにかける．

$$\mathcal{L}_\lambda^{\text{MCE-F}} = 1 - \frac{(\gamma^2+1)\cdot TP_l}{\gamma^2\cdot FN_l + FP_l + (\gamma^2+1)\cdot TP_l} \quad (9)$$

損失関数なので，また，F 値の定義域は [0-1] なので，1 から引く形で定義したが，本質的には式 (1) そのものである．$TP_l$，$FP_l$，$FN_l$ は F 値を計算する際と同様に 0-1 損失の総和と考えると以下のようになる．

$$TP_l = \sum_k \sum_{i=1}^{|s^{y^{*k}}|}\left(1 - l(d(s_i^{y^{*k}}, x, ;\lambda))\right)\cdot\delta(\mathcal{C}(s_i^{y^*}))$$

$$FP_l = \sum_k \sum_{s_i^{\hat{y}^k}\notin s^{y^{*k}}} l(d(s_i^{\hat{y}^k}, x, ;\lambda))\cdot\delta(\mathcal{C}(s_i^{\hat{y}}))$$

$$FN_l = \sum_k \sum_{i=1}^{|s^{y^{*k}}|} l(d(s_i^{y^{*k}}, x, ;\lambda))\cdot\delta(\mathcal{C}(s_i^{y^*}))$$

$s_i^y$ が抽出対象のセグメントの場合 (BIO の O タグ以外) は，$\delta(\mathcal{C}(s_i^y)) = 1$，それ以外は 0 とする．提案するセグメンテーション F 値損失関数では，$l()$ の範囲を [0-1] の区間であることを想定しているので，本稿では sigmoid 関数を使用することとする．次に，$FP_l$ の二つめの総和 ($\hat{y}_i^k \notin s^{y^{*k}}$) は，正解以外の抽出される可能性のある全てのセグメントに対して計算する必要がある．これは，semi-markov CRF[8] で紹介されている計算アルゴリズムを用いることで効率的に計算することができる．しかし，本稿では，より簡単な代替手法を紹介する．つまり，式 (6) の $\psi = \infty$ とする．これにより，最尤対立候補に出現するセグメントのみを対象に $FP_l$ を計算すればよくなる．また同時に $TP_l$，$FN_l$ の計算でも，最尤対立候補と正解のみを対象に計算すればよくなるため，実装上のパラメタ更新手続きが簡単化される．

学習では，目的関数 (損失) を最小化したいので，勾配 (gradient) が 0 になる点を求る．

$$\nabla\mathcal{L}_\lambda^{\text{MCE-F}} = \frac{\gamma^2+1}{Z_D(y,x,\lambda)}\cdot\nabla l^* + \frac{Z_N(y,x,\lambda)}{Z_D(y,x,\lambda)^2}\cdot(\nabla l - \nabla l^*)$$

$$\nabla l^* = \sum_k\sum_{i=1}^{|s^{y^{*k}}|}\nabla l(d(s_i^{y^{*k}}, x, ;\lambda))\cdot\delta(\mathcal{C}(s_i^{y^*}))$$

$$\nabla l = \sum_k\sum_{s_i^{\hat{y}^k}\notin s^{y^{*k}}}\nabla l(d(s_i^{\hat{y}^k}, x, ;\lambda))\cdot\delta(\mathcal{C}(s_i^{\hat{y}}))$$

$Z_N$ と $Z_D$ は，それぞれ式 (9) 第 2 項の分子と分母を表している．次に，式 (6) の勾配は以下のようになる．

$$\nabla d(y,x,\lambda) = -\nabla g^* + \sum_{\hat{y}\in\mathcal{Y}^k}\frac{\exp(g\cdot\psi)\cdot\nabla g}{Z_\lambda(x^k,\psi) - \exp(g^*\cdot\psi)} - \frac{\exp(g^*\cdot\psi)\cdot\nabla g^*}{Z_\lambda(x^k,\psi) - \exp(g^*\cdot\psi)}$$

目的関数や目的関数の勾配は，従来の ML/MAP 学習 [2] と同様に viterbi，forward-backward アルゴリズムを用いて効率的に計算することができる．また，最適化には最急降下法や準ニュートン法等の種々のアルゴリズムを用いて計算する．

誤り推定関数 $d()$ が $\psi = \infty$ の場合は，学習時に最尤対立候補と正解のみが必要となる．各時点のパラメタで最尤出力が正解の場合は A*アルゴリズムを用いて最尤対立候補を抽出する，それ以外は，最尤出力が最尤対立候補である．また，$\psi = \infty$ では非連続関数になるので，微分不可能な点が存在する．しかし，最適化には，**一般化確率的降下法 (Generalized Probabilistic Descent: GPD)** を用いることで効率的に (準) 最適解を得ることが理論的にも実験的にも実証されている [6]．

## 5 実験: 系列セグメンテーションタスク

CoNLL-2000 [5]，2003 [9] の shared task で使用された英語チャンキングと英語固有表現抽出のデータを用いて実験を行った．チャンキングデータは，学習 8,936 文 211,727 単語 (トークン)，テスト 2,012 文 47,377 単語，11 種類のチャンク情報 (NP，VP など) とチャンク以外を表す O タグの計 12 種類のセグメント情報が与えられている．同様に，固有表現抽出データは，学習 14,987 文 203,621 単語，テスト 3,684 文 46,435 単語，4 種類の固有表現 (PER,ORG,LOC,MISC) と固有表現以外の O タグの計 5 種類のセグメント情報が与えられている．

本稿の実験では，CRF のモデル上で，素性を一致させた状況で，ML/MAP 学習と提案手法による性能の比較を目的とする．ただし，ベースラインとして，系列的に SVM を用いて学習・推定を行う汎用ツール yamcha[*2]の結果も比較として示す．yamcha のハイパーパラメタとして，SVM のソフトマージン $C = 1$，多クラス分類手法に one vs. rest 法をもちいた．

MCE 学習には，ML/MAP と同様に系列全体に対する損失 (式 (7): CRF-MCE)，セグメント単位の損失 (式 (8)+ 正規化項: CRF-MCE-S)，セグメンテーション F 値損失 (式 (9) +正規化項: CRF-MCE-F) の 3 種類の目的関数を性能を比較した．それぞれ，誤り推定関数 $d()$ には式 (6) の $\psi = \infty$ を，平滑化関数 $l()$ には sigmoid 関数を用いた ($a = 1, b = 0$)．本稿で設定した MCE 学習の関数は，非連続・局所解を持つ可能性があるため，最適化には GPD[6] により最適化をおこなった．局所解に収束してい

[*2] http://chasen.org/ taku/software/yamcha/

表 1: 英語チャンキング (CoNLL-2000 shared task) データでの実験結果

| | F 値 | (Pre, Rec.) | Sent |
|---|---|---|---|
| CRF-MCE-F | 93.98 | (94.13,93.82) | 60.14 |
| CRF-MCE-S | 93.92 | (94.04,93.79) | 60.14 |
| CRF-MCE | 93.93 | (94.06,93.81) | 60.34 |
| CRF-MAP | 93.71 | (93.82,93.60) | 59.15 |
| CRF-ML | 93.19 | (93.24,93.15) | 56.26 |
| yamcha(SVM) | 93.61 | (93.53,93.69) | 58.00 |

表 2: 英語固有表現抽出 (CoNLL-2003 shared task) データでの実験結果

| | F 値 | (Pre., Rec.) | Sent |
|---|---|---|---|
| CRF-MCE-F | 84.81 | (85.97,83.68) | 78.61 |
| CRF-MCE-S | 84.24 | (84.62,83.87) | 77.71 |
| CRF-MCE | 83.92 | (84.19,83.64) | 77.61 |
| CRF-MAP | 83.79 | (84.05,83.53) | 77.39 |
| CRF-ML | 82.39 | (82.33,82.44) | 75.71 |
| yamcha(SVM) | 83.09 | (83.86,83.74) | 76.79 |

る可能性はあるが，今回はパラメタの初期値は全て 0 から学習した結果のみで比較をおこなう．ML/MAP 学習には，文献 [2] で紹介されている準ニュートン法 L-BFGS を用いて最適化をおこなった．また，MAP 学習と MCE 学習の正規化項 (パラメタの事前分布) には，$L_2$ ノルム (ガウス分布) を用いた．

チャンキングの素性には，文献 [10] で説明されている素性を用いた．固有表現抽出では，配布データに単語の表層情報 (大文字，小文字，数値，その他) の正規表現，および 1 から 4 文字の suffix, prefix を素性として追加した．また，両タスクとも素性の window size は前後 2 単語とした．ただし，CRF の学習には上記の素性の bigram 素性を追加し，SVM では 2 次の多項式カーネルを用いた．

### 5.1 結果および考察

表 1 にチャンキング，表 2 に固有表現抽出の実験結果を示す．表中の「F 値」の列はセグメンテーション F 値，「Sent」の列は文正解率による結果を表している．表が示すように，MCE に基づく学習により ML/MAP 学習による CRF より性能を向上させることができた．

一方で，チャンキングと固有表現抽出で効果の程度が若干違うことも示された．これは，チャンキングと固有表現抽出のデータの質の違いが考えられる．固有表現抽出のデータは 8 割以上が O(outside) タグであるため，目的関数を F 値にする効果が顕著に現れたと考えられる．一方，チャンキングでは，O タグは 1 割未満であるため，通常の系列全体に対する損失と F 値の評価基準に大きな差分がなかったことが原因であると考えられる．

### 5.2 ML/MAP 学習との関連性

平滑化関数 $l()$ に logisitic 損失 $\log(1+\exp(a\cdot d(y,x,\lambda)+b))$ をとり，誤り推定関数 $d()$ に $\psi=1$ で定数 $\mathcal{A}$ を消去した式 (6) を用いると，MCE 学習の損失関数は，ML 学習の損失関数と完全に一致する．

$$
\begin{aligned}
&\log\left(1+\exp(-g^*+\log(Z_\lambda-\exp(g^*)))\right)\\
&=\log\left(\frac{\exp(g^*)}{\exp(g^*)}+\exp(-g^*)\cdot(Z_\lambda-\exp(g^*))\cdot\frac{\exp(g^*)}{\exp(g^*)}\right)\\
&=\log\left(\frac{\exp(g^*)+(Z_\lambda-\exp(g^*))}{\exp(g^*)}\right)\\
&=\log(\exp(g^*)+(Z_\lambda-\exp(g^*)))-g^*\\
&=-g^*+\log(Z_\lambda)
\end{aligned}
$$

また，MCE 学習の損失関数に正規化項をつけた式 (7) を考えれば，MAP 学習と同一の式になる．故に，ML/MAP 学習は MCE 学習の一種でり，特定の条件下での MCE 学習を行っていることに他ならない．このことからも，MCE 学習は，非常に汎用的な枠組みを示しており，タスクに依存して様々に適用することができることがわかる．

本稿で提案した学習の枠組は，その他の関連する学習法とも強い関連性をもつが本稿では説明を省略する．

## 6 まとめ

本稿では，近年，タギング/チャンキング問題で良好な性能を示している CRF の学習方法に対して，誤り最小化に基づく学習方法の枠組を示した．この枠組により，対象タスクの評価基準を学習時の目的関数として導入が可能となり，系列セグメンテーションタスクで実際に性能が向上することを示した．提案手法は，素性の改良，モデルの改良に加え，学習時の目的関数の改良という，性能向上に向けての新たな軸となることが期待される．

## 参考文献


[1] Lafferty, J., McCallum, A. and Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proc. of ICML-2001*, pp. 282–289 (2001).

[2] Sha, F. and Pereira, F.: Shallow Parsing with Conditional Random Fields, *Proc. of HLT/NAACL-2003*, pp. 213–220 (2003).

[3] Kudo, T., Yamamoto, K., and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proc. of EMNLP-2004*, pp. 230–237 (2004).

[4] Sang, E. F. T. K. and De Meulder, F.: Representing text chunks, *Proc. of EACL-1999*, pp. 173–179 (1999).

[5] Sang, E. F. T. K. and Buchholz, S.: Introduction to the CoNLL-2000 Shared Task: Chunking, *Proc. of CoNLL/LLL-2000*, pp. 127–132 (2000).

[6] Katagiri, S., Lee, C. H. and Juang, B.-H.: New Discriminative Training Algorithms Based on the Generalized Descent Method, *Proc. of IEEE Workshop on Neural Networks for Signal Processing*, pp. 299–308 (1991).

[7] Juang, B. H. and Katagiri, S.: Discriminative learning for Minimum Error Classification, *IEEE Trans. on Signal Processing*, Vol. 40, No. 12, pp. 3043–3053 (1992).

[8] Sarawagi, S. and Cohen, W. W.: Semi-Markov conditional random fields for information extraction, *Proc of NIPS-2004* (2004).

[9] Sang, E. F. T. K. and De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *Proc. of CoNLL-2003* pp. 142–147 (2003).

[10] Kudoh, T. and Matsumoto, Y.: Use of Support Vector Learning for Chunk Identification, *Proc. of CoNLL/LLL-2000* (2000).