自 然 言 語 処 理 146－1
（２００１．１１．２０）

修正学習法による形態素解析

中川 哲治　　工藤 拓　　松本 裕治

奈良先端科学技術大学院大学 情報科学研究科

〒 630−0101 奈良県生駒市高山町 8916−5

{tetsu-na,taku-ku,matsu}@is.aist-nara.ac.jp

本稿では，表現力の高い二値分類器の機械学習アルゴリズムと計算量の小さい確率的モデルを組み合わせることにより，少ない計算量で高い精度を達成するための修正学習法を提案する．この手法を日本語の形態素解析と英語の品詞タグ付けに応用したところ，計算量を抑えながら高い精度を達成できた．さらに，Support Vector Machine によってコーパス中の形態素に重みをつけることで，コーパスの誤り検出を試みた．

キーワード：機械学習, システム混合, サポートベクターマシーン, 形態素解析, コーパスの誤り検出

Revision Learning Applied to Morphological Analysis

Tetsuji Nakagawa　　Taku Kudoh　　Yuji Matsumoto

Graduate School of Information Science, Nara Institute of Science and Technology

8916−5 Takayama, Ikoma, Nara 630−0101, Japan

{tetsu-na,taku-ku,matsu}@is.aist-nara.ac.jp

In this paper, we present a revision learning with high performance and small computational cost by combining a model with high generalization capacity and a model with small computational cost. We apply the method to Japanese morphological analysis and English part-of-speech tagging, and achieve high accuracy with small computational cost. Furthermore, we show that the corpus error detection is possible using the weights of training examples calculated by Support Vector Machines.

Keywords: Machine Learning, System Combination, Support Vector Machines, Morphological Analysis, Corpus Error Detection

1　はじめに

近年，コーパスの蓄積や計算機の能力の向上に伴い，確率・統計的手法に基づいた自然言語処理が活発に行なわれている．応用として，形態素解析，文書分類，単語の意味多義の解消など様々なタスクが試みられている．そのようなコーパスを利用した自然言語処理では，Hidden Markov Model[4]，決定木[9]，最大エントロピー法 [8]，変換規則に基づく誤り駆動方式 [3] などのモデルが利用されている．最近では Support Vector Machine(SVM)[12] のような高精度な機械学習アルゴリズムも使われるようになって

いる．このように，様々なタスクにおいて多くの学習モデルが試みられているが，この学習モデルの選択には一般的に表現力の高さと計算量の大きさのトレードオフの問題が存在する．例として，SVM のようなモデルは，モデルの表現力は高いが計算量が非常に大きい．一方，bigram や trigram のようなモデルは，計算量は小さいがモデルの表現力は低く，また多量の素性を扱うのが難しい．システムの精度を高めるためにできるだけ表現力の高いモデルを使いたい場合でも，計算量の問題によって現実的な時間で計算を実行できない場合がある．特に形態素解析のように，扱うデータの単位が小さく多量のデータを処理する必要がある場合，計算量の問題は深刻になる．

本稿ではこの問題に対処するために，表現力の高いモデルと計算量の小さいモデルを組み合わせた修正学習法を提案し，形態素解析への応用を試みる．

2　修正学習法

前節で述べたように，モデルの表現力と計算量の間にはトレードオフの関係が存在し，表現力の高いモデルを利用したい場合でも計算量が大き過ぎて利用できないことがある．しかし，精度の高い学習を行ないたい場合，全ての処理を表現力の高いモデルで行なうのではなく，計算量の小さいモデルで可能な学習は計算量の小さいモデルで処理し，計算量の小さいモデルでは失敗するような難しい学習のみを表現力の高いモデルに行なわせればよい．そこで，単純なモデルによる解析結果に対して，誤っているところだけを複雑なモデルを使って修正する修正学習を考える．

学習時には正例と負例のラベルが付けられたデータで学習を行ない，テスト時には与えられた事例に対して正か負かのラベルを付与するような学習モデルを二値分類器と呼ぶ．ここで，表現力は高いが計算量が大きい二値分類の機械学習アルゴリズムがあった場合に，ある事象 Z の次の事象が，A, B, C, D, E のどのクラスであるかを決定するような問題を考えてみる．このような，複数の正解候補の中から一つの正解を当てるという問題はマルチクラスの分類問題と呼ばれる．二値分類の機械学習アルゴリズムでこのような問題を解くための方法の一つとして，one-versus-rest という方法がある．この方法では，学習時に全ての訓練事例に対して，正解のクラスに対しては正例，それ以外のクラスに対しては負例として，全てのクラスに対して学習データを作成する (図 1 左)．そして，各クラスに対して一つの分類器を作成する．テスト時には，与えられた事例に対して学習時に作られた全ての分類器で分類を行ない，正のラベルと判定した分類器のクラスを正解とする．しかしながら，この方法では学習時には各クラスに対して全てのデータを使用して学習を行ない，テスト時には全てのクラスの分類器で分類を行なうため，計算量が非常に大きくなるという問題がある．そこで，この二値分類器に計算量の小さい確率的モデルを組み合わせた修正学習により学習を行なう方法を考える．学習の際は，始めに確率的モデルによってすべての正解候補に対して正解らしさの順位をつける．そして，順位が高い候補から順番に調べてゆき，もしその候補が正解でなければそれを負例として学習データに加え，もし正解であればそれを正例として学習データに加えるとともにそこで処理を打ち切り，残りの候補については調べない (図 1 右)．テスト時にも，まず確率的モデルによって正解候補のクラスに対して正解らしさの順位をつける．そして，順位が高い候補から二値分類器によって分類を行ない，もし負のラベルと判断された場合は次の候補を調べ，正と判断された場合はそれを正解とするとともにそこで処理を打ち切り，残りの候補については調べない．

このように，学習時に one-versus-rest 法では一つの事例に対して (⟨ クラスの数 ⟩ – 1) だけ負例の学習データが作成されるのに対し，修正学習を用いた場合は確率的モデルが誤った場合にのみ負例が作成されるため，学習データ量は非常に少なくなる．また，テスト時も one-versus-rest 法ではあらゆる正解候補のクラスについて二値分類器による分類を行うのに対して，修正学習法では確率的モデルで順位づけされた順番で正のラベルと判定されるまで分類を行なうだけなので，計算量を大幅に減らすことができる．

3　修正学習法による形態素解析

形態素解析は，自然言語処理において重要な基礎技術の一つであり，高速に高精度で処理を行なうことが必要とされている．この節では修正学習法を日本語の形態素解析と英語の品詞タグ付けに適用する方法について論じる．

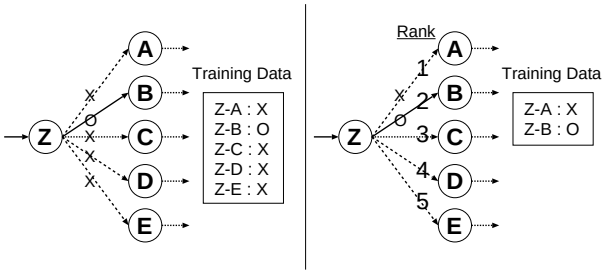

図 1: one-versus-rest 法 (左) と修正学習法 (右)

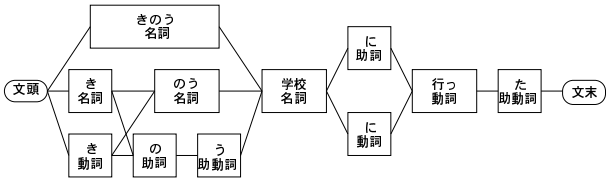

図 2: 形態素解析のラティス

3.1 ラティスの最適パス探索

日本語における形態素解析は，与えられた文に対して形態素の区切りを同定し，その品詞の付与を行なう処理である．英語における同様な処理として品詞タグ付けがあるが，英語では単語の区切りが空白によって明示的に与えられるため，品詞の付与のみを行なう．計算機で処理を行なう場合，与えられた文に対して辞書を使うことで図 2 のように，可能な形態素のつながりからなるラティスを作ることができ，この中から正解のパスを選び出すことで形態素解析を行なうことができる．ここでは，確率的モデルとして品詞 n-gram モデル [13] を，二値分類器として SVM を仮定し，それらを組み合わせた修正学習法による形態素解析を考える．n-gram モデルを用いることにより，ラティス中の任意のパスに対してそのパスのコスト (確率) を求めることができる．そこで，あるノードについて，次に接続するすべてのノードに対するコストを n-gram によって求めて順位づけをし，SVM により修正を行なう．結果として，n-gram による解析結果に対し，2 つのノードが接続するかしないかを判定するような SVM の分類器が作成される．

3.2 コーパスの誤り検出と SVM

統計的自然言語処理において各種のコーパスが利用されているが，コーパスは人手によって作成されるため，誤りを含んでいる．このような誤りは，学習時には誤った正解例を与え，テスト時には誤った評価基準を与えることになるため，コーパスに基づく自然言語処理を行なう上で大きな問題となる．そのため，コーパスの誤りを修正してその品質を高めることは重要な課題である．しかしながら，大きな規模のコーパスに対して人手で誤りを見つけ出すことは多大な労力を必要とする．そのため，自動的にコーパス中から誤り箇所を検出する技術が必要になる．

統計的処理によってコーパス中の誤りを検出する場合，一般的に誤り情報のラベルが与えられたデータは存在しないために教師なし学習の問題となるため，統計的モデルによってコーパス中で一貫性がとれていない箇所を検出することを考える．

SVM は large margin classifier の一種であり，学習データ中の各事例に対して，重みを付けることができる．また，どうしても学習できないような事例に対しては，その重みの値は上限値をとる．このように，重みの上限値を与えることで学習できない事例が存在することを許す仕組みをソフトマージンと呼ぶ．

3.1 節で述べた手法により，SVM を二値分類器として修正学習を行なった場合，正解データであるコーパス中の各形態素は全て正のラベルが与えられた事例であり，それぞれについて重みが付けられることになる．コーパス中の誤り箇所は，他の部分と比較して一貫性が無く，事例としては特殊なものであると考えられる．SVM によって学習が難しい事例や特徴的な事例には大きな重みが付けられるため，コーパス中で大きな重みが付けられた箇所を調べることによってコーパスの誤り検出が容易に行なえる可能性がある．

4 実験と考察

RWCP コーパス，京大コーパス，PennTreebank WSJ コーパスの 3 つのコーパスを用いて，修正学習法による形態素解析と品詞タグ付けの実験を行なった．

形態素解析の評価については，次のような再現率 (recall)，精度 (precision)，F 値 (F-measure) を使っ

た．

$$\text{再現率} = \frac{\text{解析結果の正解形態素数}}{\text{正解データ中の形態素数}} \quad (1)$$

$$\text{精度} = \frac{\text{解析結果の正解形態素数}}{\text{解析結果の形態素数}} \quad (2)$$

$$F\text{ 値} = \frac{2 \times \text{再現率} \times \text{精度}}{\text{再現率} + \text{精度}} \quad (3)$$

計算には Digital UNIX(Alpha 21164A 500MHz) を使用した．また SVM の計算には，SMO[7] を実装して使用した．

4.1 RWCP コーパスを用いた実験

4.1.1 実験の条件

コーパスとして，IPA 品詞体系の RWCP コーパスに話言葉等のデータが若干加えられたものを使用した．この中から学習データ 33,831 文とテストデータ 3,758 文をランダムに分けた．また，辞書として形態素解析システム茶筌の ipadic version 2.4.4 を用いた．

確率的モデルとして，品詞 bigram モデル，v-gram(可変長の n-gram) モデルによる茶筌 version 2.2.8[16] を使い，二値分類器としては 2 次の polynomial kernel の SVM を用いた．SVM の素性としては，次のものを使用した．

1. 注目している位置の形態素の単語と品詞と活用形
2. 前 2 つの形態素の単語と品詞と活用形
3. 後 2 つの形態素の単語と品詞

4.1.2 実験結果と考察

品詞 bigram と茶筌について，それぞれオリジナルの精度と修正学習後の精度の比較を行なった．形態素区切りが正しければ正解とした場合と，形態素区切りに加えて品詞の細分類も正しければ正解とした場合の結果を表 1 に示す．品詞 bigram の場合も茶筌の場合も，修正学習を適用することで F 値が改善されている．特に，形態素区切りよりも品詞付与の精度が大きく改善されている．

茶筌への修正学習の適用前と適用後の品詞ごとの F 値を，100 回以上テストデータに出現した品詞について付録の表 5 に示す．多くの品詞で F 値が改善されており，特に助詞に対する改善率が大きい．修正学習によりうまく修正が行なわれた例を図 3 に示す．*印が付けられた形態素が修正の行なわれたものである．IPA 品詞体系では，動詞「なる」にかかる状態変化を表す助詞は副詞化の助詞ではなく格助詞であり,「思う」などの推測表現の直後の「と」は引用の格助詞であると定義されているが，これらが正しく修正されている．あまり多くの素性を扱うことはできない n-gram モデルに対して，SVM では「なる」や「思う」などの前後の単語を素性として使うことができるため，多くの助詞の解析誤りに対して正しく修正が行なわれたと考えられる．

つい		副詞-一般
キッ		副詞-助詞類接続
と	*	助詞-副詞化 → 助詞-格助詞-一般
なっ		動詞-自立
て		助詞-接続助詞
しまう		動詞-非自立
誰		名詞-代名詞-一般
だろ		助動詞
う		助動詞
と	*	助詞-格助詞-一般 → 助詞-格助詞-引用
思い		動詞-自立

図 3: 修正の例

4.1.3 コーパスの誤り検出

茶筌を用いた修正学習において，学習データ 840,879 形態素の中で，SVM により重みが付けられた形態素 (support vector になったもの) は 79,872 個あり，その中で上限値の重みを持つものは 119 個あった．この中で，上限値の重みを持っていた形態素の例を図 4(上，中) に示す．*印が付けられた事例が上限値の重みを持っていた事例である．図 4(上) は，品詞の付与が誤っている．図 4(下) はコーパス中の別の文脈で現れた形態素列であるが，図 4(中) は形態素分割で一貫性がとれていない部分を検出していることが分かる．

4.2 京大コーパスを用いた実験

4.2.1 実験の条件

コーパスとして，京大コーパス version 2.0 を使用した．この中から，1 月 1 日と 1 月 3 日から 8 日までの 7 日文のデータ (7,958 文)，1 月 9 日以外の 172 日分のデータ (18,710 文) を学習データとし，1 月 9 日の 1 日分のデータ (1,246 文) をテストデータとした．辞書としては，日本語形態素解析システム JUMAN

表 1: RWCP コーパスでの実験結果

		形態素区切り			品詞細分類		
		再現率	精度	F 値	再現率	精度	F 値
品詞 bigram	オリジナル	98.06%	98.77%	98.42%	95.61%	96.30%	95.96%
	修正学習後	99.06%	99.27%	99.16%	98.13%	98.33%	98.23%
茶筌	オリジナル	99.06%	99.20%	99.13%	97.67%	97.81%	97.74%
	修正学習後	99.22%	99.34%	99.28%	98.26%	98.37%	98.32%

	する	動詞-自立
	こと	名詞-非自立-一般
	が	助詞-格助詞-一般
	確実	名詞-形容動詞語幹
*	に	助詞-副詞化
	なっ	動詞-自立
	た	助動詞
	。	記号-句点
	お互い	名詞-一般
	が	助詞-格助詞-一般
*	最低限	名詞-一般
	確保	名詞-サ変接続
	し	動詞-自立
	たい	助動詞
	お互い	名詞-一般
	が	助詞-格助詞-一般
	最低	名詞-一般
	限	名詞-接尾-一般
	確保	名詞-サ変接続
	し	動詞-自立
	たい	助動詞

図 4: コーパス誤り検出の例

version 3.61[15] に附属の辞書を使った．確率的モデルとして bigram を使い，二値分類器として 2 次の polynomial kernel の SVM を使った．SVM の素性としては，RWCP コーパスによる実験と同じものを使用した．

JUMAN による解析結果を構文解析システム KNP[14] によって曖昧性を解消したものとの比較を行なった．

4.2.2 実験結果と考察

品詞 bigram に修正学習を適用した結果を表 2 に示す．RWCP コーパスの場合と同様に，形態素分割に比べて品詞付与の改善率が大きい．また学習データの増加により，修正学習の改善率も上がっていることが分かる．

4.3 WSJ コーパスを用いた実験

4.3.1 実験の条件

コーパスとして，PennTreebank WSJ コーパスを使用した．41,342 文を学習データ，11,771 文をテストデータとしてランダムに抽出した．辞書は学習データから作成した．確率的モデルとしては ICOPOST release 0.9.0[10] の T3 を利用した．これは trigram を用いた品詞タグ付けシステムである．二値分類器としては 2 次の polynomial kernel の SVM を使った．SVM の素性としては，

1. 注目している位置の単語の 4 文字までの語頭と語尾，数字・大文字・ハイフンの有無
2. 前 2 つの単語と品詞
3. 後 2 つの単語と品詞

を使用した．

比較として，trigram による品詞タグ付けを行なう TnT[2]，one-versus-rest(1-v-r) 法による SVM を使った品詞タグ付け [6] の結果と比べた．one-versus-rest 法による SVM を使った品詞タグ付けでは，上述した修正学習法の SVM で使用するものと同じ素性を使用している．

4.3.2 実験結果と考察

WSJ コーパスで修正学習法の実験を行ない，既知語に対する精度，未知語に対する精度，全体の精度を比較したものを表 3 に示す．修正学習により，既知語，未知語ともに精度の向上が見られる．しかしながら，one-versus-rest による SVM 以上の精度は得られておらず，特に未知語に対する精度に差が出ている．この WSJ コーパスによる実験では，品詞タグ付けのための辞書を学習データから獲得している．これにより，学習時には未知語が 1 つも存在しないた

表 2: 京大コーパスでの実験結果

		形態素区切り			品詞細分類		
		再現率	精度	F 値	再現率	精度	F 値
品詞 bigram(7 日分)	オリジナル	97.72%	97.13%	97.43%	93.21%	92.65%	92.93%
	修正学習後	98.40%	97.77%	98.08%	95.62%	95.01%	95.31%
品詞 bigram(172 日分)	オリジナル	97.86%	97.20%	97.53%	93.31%	92.67%	92.99%
	修正学習後	98.67%	97.99%	98.33%	96.00%	95.34%	95.67%
JUMAN+KNP		98.89%	98.53%	98.71%	95.10%	94.75%	94.93%

めに確率的モデルは学習データの解析については未知語に対する誤りを起こさない．そのため SVM では未知語に対する修正が十分できない可能性がある．そこで，学習データを確率的モデルで解析して SVM の訓練データを作成する際に，辞書の中から出現頻度の低い単語を取り除くことで未知語に対する誤りを起こさせるようにした．1 回しか学習データ中に出現しなかった単語を辞書から取り除いた場合の結果が表 3 の “cutoff-1” である．この処理によって，未知語に対する精度を大きく改善することができている．

この修正学習法の利点の一つは計算量が少ないことであるため，確率的モデルや one-versus-rest 法との計算時間の比較を行なった．さらに，SVM の kernel 関数として linear kernel を使用した場合，モデルの表現力は低くなるが，テスト時の計算量を大きく減らすことができるため，linear kernel を使った場合の修正学習についても比較を行なった．linear kernel の SVM の計算には，SVM^{light} version 3.02[5] を使用した．学習事例数，学習時間，support vector(sv) 数，テスト時間，精度についてまとめたのが表 4 である．one-versus-rest 法では，事例に付与されたラベル以外の全てのクラスに対して負例が生成されるのに対して，修正学習では確率的モデルが誤った事例のみが負例になるため，学習事例数が少なくなり，学習時間も大きく減少していることが分かる．さらにテスト時も，one-versus-rest 法では全てのクラスに対して SVM による評価を行なうのに対して，修正学習では確率的モデルが与えた順番に正のラベルと判断されるまで評価を行なえばよいため，やはり計算時間は少なくなっている．2 次の polynomial kernel の代わりに linear kernel を使った場合の修正学習では，精度はやや落ちているものの，さらに少ない計算量で処理できることが分かる．

5 関連研究

本研究では，確率的モデルの処理結果を二値分類器により修正する方法を試みたが，これに似た方法として，Brill の変換器則に基づく誤り駆動方式がある [3]．この Brill の方法では，注目している単語の前後の単語や品詞を条件として，品詞を変更していく規則を学習し，品詞のタグ付けを行なう．本手法では，規則を使うのではなく，統計的な機械学習アルゴリズムによって品詞の修正を行なっていくという点が大きく異なる．また，本手法では始めに確率的モデルによって順番をつけ，二値分類器によってその結果が正しいか誤っているかを判定するという点でもアプローチが違っている．

複数の学習アルゴリズムを結合させることにより，高い精度を得たり計算量を削減しようとする試みは近年盛んに行なわれている [1]．複数の学習アルゴリズムを混合させる方法には，大きく分けて 2 つのアプローチがある．一つは並列的な方法 (multiexpert method) であり，もう一つは直列的な方法 (multi-stage method) である．前者は，複数の学習器でそれぞれ独立に学習，評価を行ない，全ての学習器の多数決で結果を決めるようなアプローチである．後者は，複数の学習器を順番に並べ，直前の学習器によって棄却された事例に対して学習，評価を行なっていくようなアプローチである．本手法は，後者のアプローチに属する．品詞タグ付けにおいて，前者のアプローチによる実験がいくつか行なわれており，Halteren らは Hidden Markov Model, Memory-Based, 変換器則に基づく誤り駆動方式，最大エントロピー法の 4 つのモデルに基づいた品詞タグ付けシステムで多数決を行ない，タグ付け精度を向上させたことを報告している [11]．並列的なアプローチでは各学

表 3: WSJ コーパスでの実験結果

	精度 (既知語 / 未知語)	誤りの数
T3　オリジナル	96.59% (96.90% / 82.74%)	9720
修正学習後	96.93% (97.23% / 83.55%)	8734
修正学習後 (cutoff-1)	96.98% (97.25% / 85.11%)	8588
TnT	96.62% (96.90% / 84.19%)	9626
SVM 1-v-r	97.11% (97.34% / 86.80%)	8245

表 4: WSJ コーパスを用いた計算量の比較

	学習事例数の合計	学習時間 (時間)	sv 数の合計	テスト時間 (秒)	精度
T3　オリジナル	—	—	—	89	96.59%
修正学習 ($d = 2$)	1027840	16	110563	2089	96.93%
修正学習 (linear)	1027840	2	83754	129	96.90%
SVM 1-v-r	999984×50	625	374746	55239	97.11%

習器ごとに弱点を補い合うことができ，汎化能力を高められるという利点がある一方，複数の学習器に対して全ての訓練データで学習を行ない，全てのテストデータで評価を行なうため，計算量は大きくなるという問題がある．一方，本手法のような直列的なアプローチでは，弱点を補い合うことはできないが，計算量を大きく減らすことができるため，形態素解析のように多量のデータを扱う場合や，SVM のような計算コストの大きい学習器を使う際には有効であると考えられる．

6　まとめ

本稿では，統計的モデルの表現力と計算量のトレードオフを解決するために修正学習法を提案し，日本語の形態素解析と英語の品詞タグ付けに応用した．その結果，複数のデータに対して，学習や評価に要する時間を大きく減らすとともに，高い精度を達成できることを示した．また，SVM によって学習された事例の重みを使うことでコーパスの誤り検出が行なえる見通しが得られた．

本手法は，n-gram や SVM 以外の確率的モデルや二値分類器にも適用でき，形態素解析や品詞タグ付け以外の用途にも用いることができる．また，計算量をさらに削減する方法として，確率的モデルによる順位づけにおいて，1 番目の候補が 2 番目のものに比べて十分に高い確率を持っていれば，二値分類器での評価を行なわずに正解としてしまうような改良も考えられる．

参考文献

[1] Alpaydm, E.: Techniques for Combining Multiple Learners, *Proceedings of Engineering of Intelligent Systems '98 Conference* (1998).

[2] Brants, T.: TnT — A Statistical Part-of-Speech Tagger, *Proceedings of ANLP-NAACL 2000*, pp. 224–231 (2000).

[3] Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging, *Computational Linguistics*, Vol. 21, No. 4, pp. 543–565 (1995).

[4] Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M.: Equations for Part-of-Speech Tagging, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 784–789 (1993).

[5] Joachims, T.: SVM^{light}: Support Vector Machine.
http://www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light_v3.02.eng.html.

[6] Nakagawa, T., Kudoh, T. and Matsumoto, Y.: Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines, *Proceedings of 6th Natural Language Processing Pacific Rim Symposium* (2001). (to appear).

[7] Platt, J. C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization, *Advances in kernel methods – support vector learning* (Schölkopf, B., Burges, C. and Smola, A.(eds.)), MIT Press (1998).

[8] Ratnaparkhi, A.: A Maximum Entropy Model for Part-of-Speech Tagging, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133–142 (1996).

[9] Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees, *Proceedings of the International Conference on New Methods in Language Processing*, pp. 44–49 (1994).

[10] Schröder, I.: ICOPOST — Ingo's Collection Of POS Taggers.
http://nats-www.informatik.uni-hamburg.de/~ingo/icopost/.

[11] van Halteren, H., Zavrel, J. and Daelemans, W.: Improving Accuracy in Wordclass Tagging through Combination of Machine Learning Systems, *Computational Linguistics*, Vol. 27, No. 2, pp. 199–230 (2001).

[12] Vapnik, V. N.: *The Nature of Statistical Learning Theory*, Springer, 2nd edition (1999).

[13] 永田昌明: 確率モデルによる日本語処理に関する研究, 京都大学博士学位論文 (1999).

[14] 黒橋禎夫: 日本語構文解析システム KNP version 2.0 b6 使用説明書, 京都大学大学院情報学研究科 (1998).

[15] 黒橋禎夫, 長尾眞: 日本語形態素解析システム JUMAN version 3.61, 京都大学大学院情報学研究科 (1998).

[16] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸: 形態素解析システム『茶筌』 version 2.2.8 使用説明書, 奈良先端科学技術大学院大学 松本研究室 (2001).

付録

表 5: RWCP コーパスによる茶筌の品詞ごとの F 値

品詞 (出現回数)	オリジナル	修正学習後
記号-一般 (1263)	99.33%	99.33%
記号-括弧開 (2210)	100.00%	100.00%
記号-括弧閉 (2202)	100.00%	100.00%
記号-句点 (2786)	99.95%	99.96%
記号-空白 (3007)	99.97%	99.97%
記号-読点 (4154)	99.99%	99.98%
形容詞-自立 (628)	94.12%	97.39%
助詞-格助詞-一般 (8416)	97.16%	98.17%
助詞-格助詞-引用 (757)	86.45%	94.51%
助詞-格助詞-連語 (509)	95.08%	97.66%
助詞-係助詞 (2909)	99.30%	99.38%
助詞-接続助詞 (2092)	98.65%	99.14%
助詞-副詞化 (208)	77.14%	86.00%
助詞-副助詞 (652)	97.05%	97.28%
助詞-副助詞／並立助詞 ／終助詞 (188)	97.88%	98.16%
助詞-並立助詞 (429)	87.26%	92.03%
助詞-連体化 (4053)	99.45%	99.56%
助動詞 (4419)	97.07%	98.30%
接続詞 (258)	94.74%	94.74%
接頭詞-数接続 (209)	99.29%	99.52%
接頭詞-名詞接続 (608)	94.70%	95.28%
動詞-自立 (6543)	97.78%	98.53%
動詞-接尾 (594)	99.24%	99.49%
動詞-非自立 (1068)	98.08%	98.64%
副詞-一般 (396)	92.44%	94.18%
副詞-助詞類接続 (383)	96.34%	96.73%
名詞-サ変接続 (6578)	99.42%	99.51%
名詞-一般 (12847)	97.27%	97.60%
名詞-形容動詞語幹 (776)	98.02%	98.14%
名詞-固有名詞-一般 (304)	86.27%	86.04%
名詞-固有名詞-人名-姓 (1248)	95.65%	96.69%
名詞-固有名詞-人名-名 (804)	95.13%	95.58%
名詞-固有名詞-組織 (1082)	90.40%	92.53%
名詞-固有名詞-地域-一般 (1399)	93.40%	95.05%
名詞-固有名詞-地域-国 (750)	96.48%	97.80%
名詞-数 (6482)	99.48%	99.58%
名詞-接尾-一般 (2226)	95.08%	96.01%
名詞-接尾-形容動詞語幹 (146)	98.62%	98.62%
名詞-接尾-助数詞 (2344)	99.14%	99.17%
名詞-接尾-人名 (220)	99.77%	100.00%
名詞-接尾-地域 (465)	97.71%	98.26%
名詞-接尾-副詞可能 (179)	94.79%	96.15%
名詞-代名詞-一般 (403)	98.39%	98.51%
名詞-非自立-一般 (800)	97.08%	98.06%
名詞-非自立-助動詞語幹 (124)	99.60%	99.60%
名詞-非自立-副詞可能 (448)	97.12%	97.88%
名詞-副詞可能 (1527)	98.04%	98.17%
連体詞 (378)	99.07%	99.20%