Support Vector Machineを用いた形態素解析と修正学習法の提案

中川 哲治[†] 工藤 拓[††] 松本 裕治[††]

形態素解析は自然言語処理において基本的な問題の1つである．本稿では，機械学習アルゴリズムの1つであるSupport Vector Machine（SVM）を用いて形態素解析を行う．SVMは高い表現力と性能を持つ優れた学習モデルである一方で，大きな計算コストを必要とする問題がある．そのため，一般に多量のデータと多数の分類クラスを処理する必要のある形態素解析において，妥当な時間で計算を行うことは難しい．そこで，SVMを用いて効率良く高精度に計算を行うため，表現力の高い学習モデルと計算量の小さい学習モデルを組み合わせる修正学習法を提案する．One-versus-Rest法と修正学習法によりSVMを形態素解析に適用したところ，従来手法を上回る精度が得られた．特に，修正学習法を用いることで，現実的な計算時間で高い性能を得られることを確認した．

Morphological Analysis Using Support Vector Machines and Proposal of Revision Learning

Tetsuji Nakagawa,[†] Taku Kudo[††] and Yuji Matsumoto[††]

Morphological analysis is a fundamental task in natural language processing. In this paper, we conduct morphological analysis using Support Vector Machines (SVMs). Although SVMs are excellent learning models that have high capacity and performance, they need a large computational cost. Therefore, morphological analysis, which often handle a large amount of data and a large number of classes, cannot be processed using SVMs in a reasonable time. To solve this problem, we propose a method named revision learning, which combines a learning model with high generalization capacity and a learning model with small computational cost, in order to achieve high performance with small computational cost. We applied SVMs to morphological analysis with the one-versus-rest method and the revision learning method, and obtained higher accuracies than previous works. Especially, revision learning is shown to have higher performance within a reasonable computational time.

1. はじめに

形態素解析は，自然言語処理において重要な基礎技術の1つである．情報検索や機械翻訳をはじめとした多くのアプリケーションにおいて前処理として使用されるため，高い精度が必要とされる．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う．また，形態素解析に関連した問題の1つに未知語の品詞推定がある．未知語とは，形態素解析や品詞タグ付けにおいてそのシステムの辞書中に存在しない単語であるが，未知語の品詞を精度良く推定することは難しいために，未知語の存在は形態素解析システムの性能を下げる大きな要因となっている[5]．そのため，未知語の品詞をより正確に推定することも重要な課題の1つといえる．

形態素解析や品詞タグ付けについて，多くの研究が行われている．特に最近は，人手によってルールや辞書を作成してシステムを構築する代わりに，コーパスから統計的情報やルールを学習する方法が広く用いられている．現在，日本語の形態素解析や英語の品詞タグ付けにおいて品詞n-gramモデルがしばしば用いられている[10),25]．この方法では，文中の各単語の品詞を観測不可能な状態と見なす．そして，単語の出現確率はその状態にのみ依存し，状態の遷移確率は $(n-1)$ 個前までの状態にのみ依存すると考える．このような

† 沖電気工業株式会社研究開発本部情報サービスシステムラボラトリ
Service Media Laboratory, Corporate Research & Development Center, Oki Electric Industry Co., Ltd.
†† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

仮定のもとで任意の単語列が与えられたとき，その観測不能な状態の遷移系列の候補，つまり各単語に対する品詞の候補の中から，最も尤度が高いものを Viterbi アルゴリズムにより効率的に求めることができる．この方法は，学習や解析に要する計算量が少なく比較的高い精度を持つ．しかしながら，多量の素性を扱うのは難しく，出現頻度の低い事象に対処するためにスムージングを工夫して行う必要がある．n-gram モデルのほかにも，決定木[20] や変換規則に基づく誤り駆動モデル[6]，最大エントロピー法[18] 等を用いた方法が提案されているが，使用する素性の組合せを人手で行ったりスムージングの方法を工夫する等様々なチューニングを必要とすることが多い．

本稿では機械学習アルゴリズムの 1 つである Support Vector Machine（SVM）を用いた形態素解析を試みる．SVM は近年提案された機械学習のアルゴリズムで，多量の素性を用いても過学習を起こしにくく，またカーネル関数を利用することでモデルの表現力を柔軟に変えられる等の特徴を持っている．そのため，SVM を用いて形態素解析を行うことで，特別なチューニングを行わなくても十分高い精度を得られる可能性がある．しかしながら，SVM は優れた性能を持つ反面，大きな計算量を要するという問題がある．一般に，形態素解析では多量のデータを扱わなくてはならないため，計算量の問題は非常に深刻なものとなる．そこで本稿では，計算量の問題に対処するために SVM を他の計算量の小さい学習モデルと組み合わせた修正学習法を提案し，形態素解析に適用する．以下，2 章では SVM について説明を行い，3 章では SVM を直接用いた未知語の品詞推定と品詞タグ付けについて述べる．4 章では，計算量の問題に対処するための修正学習法を導入し，品詞タグ付けと形態素解析に適用する．最後に 5 章で結論を述べる．

2. Support Vector Machine

Support Vector Machine（SVM）は二値分類のための機械学習モデルである[23]．L 次元の素性ベクトル $\mathbf{x} \in \mathbf{R}^L$ に対し，

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \qquad \mathbf{w} \in \mathbf{R}^L,\ b \in \mathbf{R}, \tag{1}$$

で表される分離平面を考え，$+1$（正例）か -1（負例）の値をとる y_i をラベルとして持つ訓練事例 $\mathbf{x}_i$ を次のように分類するとする：

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \tag{2}$$

この不等式を満たす分離平面は一般に無数に存在するが，SVM では分離平面とそれに最も近い事例間の距離（マージン）を最大にするような $\mathbf{w}$ と b を求める．このような分離平面は分類誤りの期待値を最小にすることが知られており，これによって SVM は素性の次元数に影響されずに高い汎化能力を持っている．テスト事例 $\mathbf{x}$ が与えられた場合，次のような判別関数 $f(\mathbf{x})$ を使い，この値の正負によって事例のラベル y を決定する：

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{3}$$

$$y = \mathrm{sgn}(f(\mathbf{x})). \tag{4}$$

式 (1) の分離平面では与えられた事例を線形分離することしかできない．そこで線形分離不可能な場合を扱うために，素性ベクトルをより高次元の空間へ非線形関数により写像し，そこで線形分離を行うことを考える．SVM におけるモデルの学習や事例の分類には，素性ベクトルどうしの内積しか使わないという性質があるため，2 つの素性ベクトルに対し，それらの高次元空間上における写像の内積を与える関数が存在すれば，高次元空間を意識せずに計算を行うことができる．このような関数をカーネル関数と呼び，カーネル関数を使用することで素性ベクトルを直接高次元空間へ写像することなく効率的に計算を行うことができる．カーネル関数を $K(\mathbf{x}_i, \mathbf{x}_j)$ とすると，この場合の判別関数 $f(\mathbf{x})$ は次のように表される．

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{5}$$

ここで，α_i は各訓練事例に対して与えられる係数であるが，最適な分離平面を表す α_i は次の二次計画問題を解くことによって求められる．

$$\min.\ \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i, \tag{6}$$

$$\text{s.t.}\ 0 \leq \alpha_i \leq C, \tag{7}$$

$$\sum_i \alpha_i y_i = 0. \tag{8}$$

上の式では，α_i に対して定数 C 以下であるような制約が加えられているが，これはすべての訓練事例を完全に線形分離できない場合を許容するためのものであり，ソフトマージンと呼ばれる手法である[23]．

カーネル関数として様々なものが提案されており，その 1 つに次のように定義される多項式カーネルがある：

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d. \tag{9}$$

これは，各素性の d 個までの組合せを考えた高次元空間上で線形分離を行うことに相当する．つまり，この多項式カーネルを用いることで素性の組合せを自動的に考慮することができる．この多項式カーネルは設定するパラメータの数が少なく，またテキスト分類[29]

やチャンキング[15] 等の自然言語処理に広く用いられているため，本稿でもこのカーネル関数を使用することにする．また，ソフトマージンの定数 C については，値を変えた場合でも結果に対する大きな差は見られなかったため，本稿での実験はすべて $C = 1$ として行う．

3. SVM を直接用いた未知語の品詞推定と品詞タグ付け

本章では，前章で説明した SVM を直接利用して，英語における未知語の品詞推定と品詞タグ付けを行う．

3.1 One-versus-Rest 法による未知語の品詞推定

品詞タグ付けの後処理として行う場合の未知語の品詞推定は，与えられた未知語に対して複数個の品詞（クラス）候補の中から正解の品詞（クラス）を 1 つ選ぶ多値分類問題と考えることができる．しかしながら，SVM は二値分類器であるため，多値分類問題を直接扱うことはできない．二値分類器で多値分類問題を扱う方法は様々なものが提案されているが[1]，従来研究において各手法の違いによる一貫した差は見られないため，ここではよく知られた方法の 1 つである One-versus-Rest 法を利用する．

One-versus-Rest 法は，k 個のクラスが存在した場合，学習時に 1 つのクラスとそれ以外の $(k-1)$ 個のクラスとを分類するような二値分類器 $f_i(\mathbf{x})\,(0 \le i \le k-1)$ を各クラスに対して作成する：

$$\begin{cases} f_i(\mathbf{x}) \ge +1 & \mathbf{x} \text{ がクラス } i \text{ に属するとき,} \\ f_i(\mathbf{x}) \le -1 & \text{それ以外.} \end{cases} \tag{10}$$

テスト時は，テスト事例に対する判別関数の値（分離平面からの距離）が最も大きくなるようなクラス c を選択する：

$$c = \operatorname*{argmax}_i f_i(\mathbf{x}). \tag{11}$$

このアルゴリズムを疑似コードで表すと，**図 1** のようになる．

```
# One-versus-Rest 法による学習
# 訓練事例 {(x_i, y_i)} を入力とし，二値分類器の学習を
# 行う．
#    C = {c_0, ..., c_{k-1}}:  クラスの集合,
#    x_i:  i 番目の学習事例,
#    y_i ∈ C:  x_i の属するクラス,
#    k:  クラスの数,
#    l:  訓練事例の数,
#    f_j(·):  クラス c_j に対する二値分類器.
procedure Train_OVR({(x_0, y_0), ..., (x_{l-1}, y_{l-1})})
begin
  # 多値分類問題を二値分類問題に変換する
  for i := 0 to l − 1
  begin
    for j := 0 to k − 1
    begin
      if c_j ≠ y_i then
        x_i を負例としてクラス c_j の訓練データに追加.
      else
        x_i を正例としてクラス c_j の訓練データに追加.
    end
  end
  # 二値分類器の学習を行う
  for j := 0 to k − 1
    作成された訓練データを使い分類器 f_j(·) の学習を
    行う.
end

# One-versus-Rest 法による評価
# テスト事例を入力とし，そのクラスを予想して返す
#    C = {c_0, ..., c_{k-1}}:  クラスの集合,
#    x:  テスト事例,
#    k:  クラスの数,
#    f_j(·):  上の手続きにより学習された二値分類器.
function Test_OVR(x)
begin
  for j := 0 to k − 1
    confidence_j := f_j(x)
  return c_{argmax_j confidence_j}
end
```

図 1 One-versus-Rest 法のアルゴリズム
Fig. 1 Algorithm of One-versus-Rest.

このアルゴリズムを疑似コードで表すと，**図 1** のようになる．

未知語の品詞推定には以下の 3 種類の素性を使用する：

品詞 未知語の前後 2 つの品詞.

単語 未知語の前後 2 つの単語.

部分文字列 未知語の 4 文字以下の語頭と語尾，未知語が数字・大文字・ハイフンをそれぞれ含むかどうか.

これは Ratnaparkhi[18] が最大エントロピー法を用いて英語の品詞タグ付けと未知語の品詞推定を行った際に用いたものとほぼ同じものである．例として，

`she/PRP returned/VBD to/TO Greenville/`未知語
`two/CD days/NNS before/IN`

という文脈を考える．この場合の未知語 `Greenville` に対する素性は，**表 1** のようになる．このような素性を使い，訓練データ中のすべての単語を訓練事例として各品詞に対する SVM の分類器を作り，未知語の品詞推定を行う．

表 1 未知語の品詞推定における素性の例
Table 1 Example of features for unknown word guessing.

品詞	t_{-1} =`TO`, t_{-2} =`VBD`, t_{+1} =`CD`, t_{+2} =`NNS`
単語	w_{-1} =`to`, w_{-2} =`returned`, w_{+1} =`two`, w_{+2} =`days`
部分文字列	`^g`, `^gr`, `^gre`, `^gree`, `e$`, `le$`, `lle$`, `ille$`, 〈大文字含む〉

3.2 One-versus-Rest 法による品詞タグ付け

未知語の品詞推定では，前後の品詞が既知なものとして SVM により単語の品詞を求めた．これを一般化して，与えられた文中のすべての単語の品詞を求めることを考えれば，品詞のタグ付けを行うことができる．

しかしながら，品詞タグ付けの後処理としての未知語の品詞推定の場合，前後の単語の品詞は既知なためこれらを素性として使用できたが，品詞タグ付けの場合は後方の単語の品詞情報を直接利用することができない．そこで，ここでは以下のような 2 つの方法を試みた．

3.2.1 前方の品詞のみを使用する方法

1 つ目の方法では，注目している単語の前方の品詞情報のみを使用する．素性として，次のものを使用することになる：

品詞 未知語の前 2 つの品詞.

単語 未知語の前後 2 つの単語.

部分文字列 未知語の 4 文字以下の語頭と語尾，未知語が数字・大文字・ハイフンをそれぞれ含むかどうか.

解析時の解の探索に関して，品詞 n-gram 等の確率的なモデルでは，すべての可能な品詞系列の中から最尤のものを Viterbi アルゴリズムにより効率的に求めることができる．一方，SVM は直接確率値を扱えないため，同様の方法を使うことができない．そこで，ここでは文頭の単語から 1 語ずつ順番に品詞を決定していく方法をとることにした．このような決定的な方法は，局所的な情報しか考慮せず全体的な尤度を考慮しないという問題があるが，計算量は少なくてすむという利点がある．

解析時には，訓練データ中に現れた単語に対しては効率良く計算するために辞書を用いて可能な品詞の候補を与え，未知語に対してはすべての品詞を候補として与えた．この点を除けば，単語の出力確率等は使用せず，部分文字列や前後の単語しか利用していないため，未知語に対する特別な処理は行わずに解析を行うことができる．

3.2.2 前後の品詞を使用する方法

もう 1 つの方法では，前後の品詞を使用する．後方の品詞も素性とする場合，解析中の単語よりも後にある単語の品詞を利用することになるが，それらは一般に未知である．Roth ら[19] は，訓練データ中に現れた各単語に対して最も多かった品詞を辞書に記録しておき，その品詞を各単語に対する一時的な品詞として初めに付与しておくことで，前後の品詞を素性として用いた品詞タグ付けを行っている．しかし，この方法は未知語が存在する場合に適用することができない．ここでは，それと似た次のような辞書を用いない 2 つのパスからなる方法を用いた．まず 1 パス目では後方の品詞情報を使用せずに，3.2.1 項と同じ方法で各単語の品詞を推定する．2 パス目では，後方の単語の品詞については 1 パス目で推定した品詞を用いて，表 1 と同じ素性を使って文の先頭から品詞を再推定する．この方法も，やはり未知語に対する特別な処理は必要としない．

3.3 実験と考察

Penn Treebank WSJ コーパス（品詞の数は 50）を使って未知語の品詞推定と品詞タグ付けの実験を行った．訓練データは文単位でランダムに抽出し，約 1,000，10,000，100,000，1,000,000 トークンのものを用意した．

未知語の品詞推定のためのテストデータは，訓練データに現れなかった単語を使用した．また，未知語の前後の品詞は品詞タグ付けシステム TnT[5] によってタグ付けした．

品詞タグ付けのためのテストデータは，コーパスの全データから訓練データを除いた残りの約 285,000 トークンを使用した．

未知語の品詞推定と品詞タグ付けにおいて，品詞タグ付けシステム TnT との比較を行った．これは，品詞 trigram モデルに基づいて品詞タグ付けを行うシステムで，未知語の品詞推定には，異なる長さの語尾に対する条件付き確率を線形補間したものを使用している．

3.3.1 未知語の品詞推定の実験

訓練データ量を変えた場合の未知語に対する品詞推定の結果を**表 2** に示す．ここで，“SVM” は本手法による精度，“d” はそのとき使用した多項式カーネルの次数，“TnT” は TnT による精度を表す．すべての訓練データ量に対して，提案手法は TnT よりも高い精度を得ている．

SVM で使用する素性等の条件を変えた場合の結果を**表 3** に示す．左から順に，未知語の前後の品詞に正しいものを与えた場合，そして前後の品詞，前後の単語，未知語の部分文字列のそれぞれの素性を使わなかった場合を表す．表 2 の実験では未知語の前後の品詞を TnT によって付与したが，その際の誤りによって未知語の品詞推定精度が 1%ほど低下していたことが分かる．また，部分文字列の情報が未知語の品詞推

表 2 訓練トークン数に対する未知語の品詞推定精度
Table 2 Performance of unknown word guessing vs. number of training tokens.

訓練トークン数	SVM	d	TnT
1,000	69.8%	1	69.4%
10,000	82.3%	2	81.5%
100,000	86.7%	2	83.3%
1,000,000	87.1%	2	84.2%

定に最も寄与しており，続いて前後の品詞，前後の単語の順に有用であることが分かる．

一般に，訓練データ中に少数しか出現しない素性は統計的に十分信頼できず，しばしばシステムの性能を低下させる．そのため，Ratnaparkhi[18] は最大エントロピー法を用いた品詞タグ付けにおいて，10 回未満しか出現しなかった素性は無視することでこの問題に対処している．そこで，我々も低頻度の素性を削除した場合の精度を 10,000 トークンの訓練データを使った場合について調べた結果，**表 4** の結果が得られた．この結果を見ると，素性を削減することによって多少精度は上がっているが，このようなヒューリスティックなチューニングを行わなくても SVM は過学習を起こしにくく，性能はあまり低下していないことが分かる．

多項式カーネルの次数を変えた場合の結果を**表 5** に示す．このタスクでは，次数が 2 の場合に最も高い精度が得られている．

表 3 素性を変えた場合の未知語の品詞推定精度
Table 3 Performance of unknown word guessing for different sets of features.

訓練トークン数	品詞正解	品詞なし	単語なし	部分文字列なし
1,000	71.1%	64.2%	68.7%	33.7%
10,000	82.9%	75.8%	80.2%	37.1%
100,000	87.5%	80.1%	85.2%	33.8%
1,000,000	88.5%	80.8%	86.0%	30.0%

表 4 素性を削減した場合の未知語の品詞推定精度：“cutoff” 以下の回数しか出現しなかった素性は無視した場合
Table 4 Performance of unknown word guessing along reduction of features: features occurring less than or equal to “cutoff” are not taken into consideration.

Cutoff	0	1	2	3	4
精度	82.3%	82.5%	82.6%	82.6%	82.5%
素性の数	18,936	7,683	4,854	3,493	2,792
	10				
	81.7%				
	1,314				

表 5 カーネル関数を変えた場合の未知語の品詞推定精度
Table 5 Performance of unknown word guessing for different kernel functions.

訓練トークン数	多項式カーネルの次数			
	1	2	3	4
1,000	69.8%	69.8%	61.2%	34.8%
10,000	82.0%	82.3%	80.3%	79.5%
100,000	85.0%	86.7%	86.0%	84.3%
1,000,000	85.2%	87.1%	—	—

（“—” で示す欄は計算時間が長くて結果が出せなかったもの）

3.3.2 品詞タグ付けの実験

品詞タグ付けの実験結果を**表 6** に示す．前方の品詞のみを使用した場合と，前後の品詞を使用した場合の結果を示しているが，前者は TnT と同等の性能であり，後者はそれを上回っている．これらの 2 つの方法は既知語に比べて未知語に対する精度に大きな差がみられる．

前後の品詞を使用する方法で素性を変えた場合の結果を**表 7** に示す．各素性の精度に対する寄与の度合いは，未知語の品詞推定の場合とほぼ同じ傾向を持っている．

3.4 問 題 点

One-versus-Rest 法により SVM を直接的に用いて未知語の品詞推定と品詞タグ付けを行った結果，従来手法を上回る高い精度を得ることができた．しかしながら，本手法には次のような 2 つの大きな問題がある：

(1) 計算量が大きい．

表 6 品詞タグ付けの精度（既知語/未知語）
Table 6 Performance of POS tagging (for known words/for unknown words).

訓練トークン数	SVM				TnT
	前の品詞	d	前後の品詞	d	
1,000	83.4%(96.3/68.3)	1	83.9%(96.4/69.3)	1	83.8%(96.0/69.4)
10,000	92.1%(95.5/81.2)	2	92.5%(95.7/82.2)	2	92.3%(95.7/81.5)
100,000	95.6%(96.5/85.7)	2	95.9%(96.7/86.7)	2	95.4%(96.4/83.3)
1,000,000	97.0%(97.3/86.3)	2	97.1%(97.3/86.8)	2	96.6%(96.9/84.2)

表 7 素性を変えた場合の品詞タグ付けの精度（既知語/未知語）
Table 7 Performance of POS tagging for different sets of features (for known words/for unknown words).

訓練トークン数	品詞なし	単語なし	部分文字列なし
1,000	81.3%(96.0/64.2)	83.2%(96.2/68.0)	65.2%(96.2/29.0)
10,000	90.3%(94.7/75.8)	91.9%(95.5/79.9)	80.7%(94.8/34.4)
100,000	93.9%(95.1/80.1)	95.4%(96.4/85.0)	82.4%(87.0/30.7)
1,000,000	95.0%(95.3/80.8)	96.7%(96.9/85.7)	74.4%(75.6/24.1)

(2) 日本語形態素解析に適用できない．

1つの問題は計算量である．前後の品詞を利用した品詞タグ付けにおいて，訓練データ数が 100,000 トークンの場合，学習に 17 時間，解析に 1 文あたり 1.2 秒を要した☆．さらに訓練データ数が 1,000,000 トークンの場合，学習に 26 日，解析に 1 文あたり 4.7 秒という膨大な計算時間を必要とした．SVM はもともと（訓練・評価ともに）計算コストの大きい学習アルゴリズムであるが，One-versus-Rest 法を用いる場合にはクラスの数だけ分類器を用意するため，さらに必要とされる計算量が大きくなる．規模の小さい文書分類等のタスクではそれでも妥当な時間で計算を行うことができるが，形態素解析のようなタスクでは，処理の対象となる単位が小さく，多量の事例や多数のクラスを扱うため，現実的な時間で計算を行うことは困難である．別のもう1つの問題は，日本語の形態素解析を直接行うことができない点である．日本語の形態素解析では，英語の場合のような単語の品詞の決定だけではなく，単語の区切りの同定も考えなければいけない．そのため，単純な多値分類問題として扱うことができず，ここで用いた方法をそのまま適用することはできない．

4. 修正学習法により SVM を用いた品詞タグ付けと形態素解析

前章では，One-versus-Rest 法により SVM を用いて未知語の品詞推定と品詞タグ付けを行った．その結果，計算量と日本語形態素解析への適用に関して問題があることをみた．本章では，計算量の問題を解決するために，表現力の高いモデルに対して計算量の小さいモデルを組み合わせることによって計算コストを抑えながら高い精度を達成する，修正学習法を提案する．そして，これを日本語の形態素解析へ適用し，実験を行う．

4.1 修正学習法

SVM は高い表現力を持つが，必要とする計算量も多い．一方で，従来から使われている品詞 n-gram モデルは，SVM ほどモデルの表現力は高くないものの，品詞タグ付けを 95%以上の精度で行うことができ[5]，必要とする計算量は小さい．学習モデルに関するこのような表現力と計算量のトレードオフを考えると，品詞 n-gram のような表現力の低いモデルでも大部分の処理を行うには十分であり，SVM のような表現力の高いモデルを使ってすべての処理を行うのは非常に無駄が多いといえる．つまり，精度の高い学習を行いたい場合，すべての処理を表現力の高いモデルで行うのではなく，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ような修正学習法を考える．

One-versus-Rest 法における訓練時には，1つの訓練事例に対してすべての候補のクラスについて正例，あるいは負例を作成した．また評価時には，1つのテスト事例に対して，すべてのクラスの二値分類器で分類を行いそのクラスを決定した．修正学習法における訓練時には，1つの訓練事例に対して初めに確率的モデルによってすべての正解候補のクラスについて正解らしさの順位を付ける．ただしこの確率的モデルは，表現力は高くなくてもよいが計算量が小さいモデルであるとする．そして，順位が高いクラスから順番にその事例のラベルと等しいかどうか調べていく．もしそのクラスが正解でなければそれを負例として訓練データに加え，もし正解であればそれを正例として訓練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．このようにして作成された 2 クラスの訓練データに対して，二値分類器の学習を行う．ただしこの二値分類器は，計算量は大きくてもよいが表現力の高いモデルであるとする．評価時も，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付ける．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．

例として，ある事例 x が，A，B，C，D，E の 5 つのクラスを候補として持つ多値分類問題を考える．ただし，確率的モデルは A，B，C，D，E の順番に正解らしさの順位を与えるとし，x のクラスは B であるとする．このとき，One-versus-Rest 法と修正学習法により生成される二値の訓練データはそれぞれ**図 2** のようになり，前者では 5 つの訓練データが作成されるのに対し，後者では 2 つしか生成されない．このように，学習時に One-versus-Rest 法では 1 つの事例に対して (⟨ クラスの数 ⟩ − 1) だけ負例の訓練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

☆ 計算には Digital UNIX（Alpha 21164A 500 MHz）を使用した．

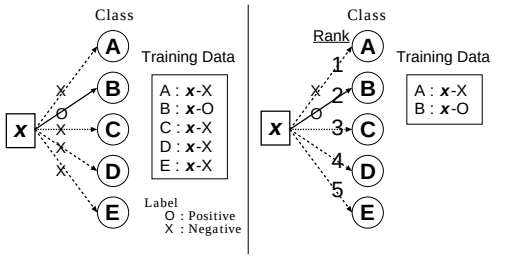

図 2 One-versus-Rest 法（左）と修正学習法（右）
Fig. 2 One-versus-Rest method (left) and revision learning method (right).

```
# 修正学習法による学習
# 訓練事例 {(x_i, y_i)} を入力とし，二値分類器の学習を
# 行う.
#   C = {c_0, ..., c_{k-1}}:  クラスの集合,
#   x_i:  i 番目の訓練事例,
#   y_i ∈ C:  x_i のクラス,
#   k:  クラスの数,
#   l:  訓練事例の数,
#   n_i:  順位付けされた C の要素の添字,
#   f_j(·):  クラス c_j に対する二値分類器.
procedure Train_RL({(x_0, y_0), ..., (x_{l-1}, y_{l-1})})
begin
  # 多値分類問題を二値分類問題に変換する
  for i := 0 to l − 1
  begin
    確率的モデルを呼び出し，次のような順位付けされた
    添字 {n_0, ..., n_{k-1}} を得る：
     P(c_{n_0}|x_i) ≥ ··· ≥ P(c_{n_{k-1}}|x_i).
    for j := 0 to k − 1
    begin
      if c_{n_j} ≠ y_i then
        x_i を負例として c_{n_j} の訓練データに追加する.
      else
      begin
        x_i を正例として c_{n_j} の訓練データに追加する.
        break
      end
    end
  end
  # 二値分類器の学習を行う
  for j := 0 to k − 1
    作成された訓練データを使い分類器 f_j(·) の学習を
    行う.
end

# 修正学習法による評価
# テスト事例を入力とし，その予想されたクラスを返す
#   C = {c_0, ..., c_{k-1}}:  クラスの集合,
#   x:  テスト事例,
#   k:  クラスの数,
#   n_i:  順位付けされた C の要素の添字,
#   f_j(·):  上の手続きにより学習された二値分類器.
function Test_RL(x)
begin
  確率的モデルを呼び出して，次のような順位付けされた
  添字 {n_0, ..., n_{k-1}} を得る：
   P(c_{n_0}|x) ≥ ··· ≥ P(c_{n_{k-1}}|x).
  for j := 0 to k − 1
    if f_{n_j}(x) > 0 then
      return c_{n_j}
  # 二値分類器が答を出さなかった際は，確率的モデルが
  # 最も尤度が高いと判断したクラスを答とする.
  return c_{n_0}
end
```

図 3 修正学習法のアルゴリズム
Fig. 3 Algorithm of revision learning.

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付けされた順に正のラベルと判定されるまで分類を行うだけなので，計算量を大幅に減らすことができる．この修正学習法のアルゴリズムを疑似コードで表現したものを**図 3** に示す．

4.2 修正学習法による形態素解析と品詞タグ付け

前節では，多値分類問題を解くための修正学習法を説明したが，これを日本語の形態素解析に適用する方法を考える．

計算機で形態素解析を行う場合，ある与えられた文に対して，形態素辞書を使うことで**図 4** のように，すべての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る．この中から最も尤度の高いパスを選び出すことで形態素解析を行うことができる．ここでは，修正学習で使用する確率的モデルとして品詞 n-gram モデル[25] を，二値分類器として SVM を考える．n-gram を用いることにより，ラティス中の文頭から始まる任意の部分パスに対してその生起確率を求めることができる．そこで，次のような手続きにより形態素解析を行う．まず文末のノードに注目する．次に，現在注目しているノードに接続するすべての前方のノードに対する確率を n-gram によって求めて順位付けし，SVM によりそのうちの 1 つを選択する，という修正学習を行う．そして，その選択されたノードに注目して同様の処理を文頭のノードまで繰り返す．ここで SVM は，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していることになる．

SVM では以下の素性を使用することにする：

(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☆

文末から解析を進めるため，前方の形態素の単語や品詞は未知であるが，n-gram によって最も尤度の高いものを推定することができるので，それを SVM の素性に使用する．

英語の品詞タグ付けは，あらかじめ単語の区切りが行われていると考えれば，日本語の形態素解析の特別な場合と見なすことができるため，上述の方法と同様に解析を行える．英語の品詞タグ付けにおいては，以下の素性を使用した

☆ 日本語では，活用形は直後にくる形態素に大きく影響するため，注目している位置と前方の形態素に関してのみ活用形を素性として用いた．

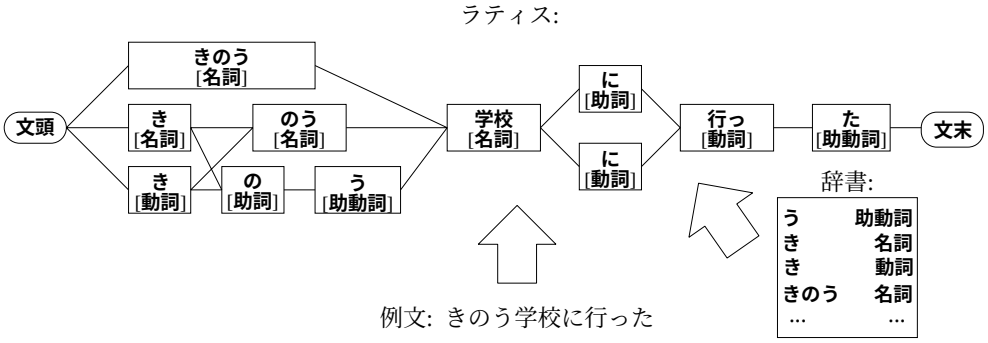

図 4 日本語形態素解析のラティスの例
Fig. 4 Example of lattice for Japanese morphological analysis.

表 8 WSJ コーパスでの実験結果
Table 8 Result on the WSJ corpus.

		精度 (既知語 / 未知語)	誤りの数
T3	オリジナル	96.59% (96.90% / 82.74%)	9,720
	修正学習後	96.93% (97.23% / 83.55%)	8,734
	修正学習後 (cutoff-1)	96.98% (97.25% / 85.11%)	8,588
TnT		96.62% (96.90% / 84.19%)	9,626
SVM One-versus-Rest		97.11% (97.34% / 86.80%)	8,245

(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) （n-gram によって推定された）前 2 つの単語と品詞
(3) 後 2 つの単語と品詞

4.3 実験と考察

Penn Treebank WSJ コーパス，RWCP コーパス，京大コーパスの 3 つのコーパスを用いて，修正学習法による英語品詞タグ付けと日本語形態素解析の実験を行った．

形態素解析の評価には，次のような再現率（recall），精度（precision），F 値（F-measure）を使用した．

$$\text{再現率} = \frac{\text{解析結果の正解形態素数}}{\text{正解データ中の形態素数}}, \tag{12}$$

$$\text{精度} = \frac{\text{解析結果の正解形態素数}}{\text{解析結果の形態素数}}, \tag{13}$$

$$\text{F 値} = \frac{2 \times \text{再現率} \times \text{精度}}{\text{再現率} + \text{精度}}. \tag{14}$$

4.3.1 品詞タグ付け

コーパスとして，Penn Treebank WSJ コーパス（品詞の数は 50）を使用した．3.3 節の品詞タグ付けの実験で使用したものと同じ，1,000,000 トークンの訓練データ（41,342 文）と，285,000 トークンのテストデータ（11,771 文）を使った．辞書は訓練データから作成した．確率的モデルとしては ICOPOST release 0.9.0 [21] の T3 を利用した．これは品詞 trigram モデルによる品詞タグ付けシステムである．二値分類器としては 2 次の多項式カーネルの SVM を使った．また，3 章で説明した One-versus-Rest 法により SVM を適用した品詞タグ付けと TnT の結果との比較を行った．

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 8** 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 法による 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．この実験では，品詞タグ付けのための辞書を訓練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 つも存在しないために確率的モデルは訓練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，訓練データを確率的モデルで解析して SVM の二値訓練用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 回しか訓練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 8 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 1 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った．また，SVM のカーネル関数として線形カーネル ($K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$) を使用

表 9 WSJ コーパスでの計算量の比較
Table 9 Computational cost on the WSJ corpus.

		総学習事例数	学習時間 (時間)	テスト時間 (秒/文)	精度
T3	オリジナル	—	0.004	0.0076	96.59%
	修正学習後 (多項式)	1,027,840	16	0.18	96.98%
	修正学習後 (線形)	1,027,840	2	0.011	96.94%
SVM One-versus-Rest		49,999,200	625	4.7	97.11%

表 10 RWCP コーパスでの実験結果
Table 10 Result on the RWCP corpus.

		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%

した場合についても調べ，比較を行った．線形カーネルを使用した場合，モデルの表現力は高くないが，テストデータに対する判別を行う際に 2 章の式 (5) の代わりに式 (3) のようにあらかじめ係数を計算しておくことができるため，評価時の計算量を大きく減らすことができる．学習事例数，学習時間，テスト時間，精度について結果をまとめたのが**表 9** である．One-versus-Rest 法では，事例に付与されたラベル以外のすべ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．さらに評価時も，One-versus-Rest 法ではすべてのクラスに対して SVM による評価を行うのに対して，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 次の多項式カーネルの代わりに線形カーネル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ていることが分かる．

4.3.2 RWCP コーパスでの形態素解析

コーパスとして，IPA 品詞体系の RWCP コーパスに話し言葉等のデータが若干加えられたものを使用した（品詞の数は 88）．この中から訓練データ 33,831 文とテストデータ 3,758 文をランダムに取り出して使用した．また，辞書には形態素解析システム茶筌の ipadic version 2.4.4 [31] を用いた．

確率的モデルとして，可変長 n-gram モデルと品詞 bigram モデルを使い，二値分類器としては 2 次の多項式カーネルの SVM を用いた．なお，可変長 n-gram モデルとしては，実際には茶筌 version 2.2.8 [4),30] に冗長解析を行わせた結果を利用した．

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 10** に示す．品詞 bigram の場合も茶筌の場合も，修正学習法を適用することで再現率，精度，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．

修正学習の適用前と適用後の茶筌の出力において，正しく品詞が付与された形態素の数を大分類の品詞ごとにまとめたものを**表 11** に示す．修正学習により多くの品詞で正しく修正が行われているが，特に助詞に対する改善が大きい．修正学習によりうまく修正が行われた例を**表 12** に示す．IPA 品詞体系では，動詞「なる」に係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」等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」等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われたと考えられる．

4.3.3 京大コーパスでの形態素解析

コーパスとして，京大コーパス version 2.0 を使用した（品詞の数は 43）．この中から，1 月 1 日と 1 月 3 日から 8 日までの 7 日分のデータ（7,958 文），1 月 9 日以外の 172 日分のデータ（18,710 文）を訓練データとし，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

表 11 正しく品詞が付与された形態素の数
Table 11 The number of correctly tagged morphemes for each POS category tag.

品詞	出現回数	オリジナル	修正学習後	差
名詞	41,512	40,355	40,556	+201
接頭詞	817	781	784	+3
動詞	8,205	8,076	8,115	+39
形容詞	678	632	655	+23
副詞	779	735	750	+15
連体詞	378	373	373	0
接続詞	258	243	243	0
助詞	20,298	19,686	19,942	+256
助動詞	4,419	4,333	4,336	+3
感動詞	94	90	91	+1
記号	15,665	15,647	15,651	+4
その他	1	1	1	0
フィラー	43	36	36	0

表 12 RWCP コーパスにおける修正された文の例
Table 12 Example sentences of revisions on the RWCP corpus.

	単語	品詞		
			オリジナル	修正後
例 1	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	
例 2	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	

表 13 京大コーパスでの実験結果
Table 13 Result on the Kyoto University corpus.

		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	97.72%	97.13%	97.43%	93.21%	92.65%	92.93%
(7,958 文)	修正学習後	98.40%	97.77%	98.08%	95.62%	95.01%	95.31%
品詞 bigram	オリジナル	97.86%	97.20%	97.53%	93.31%	92.67%	92.99%
(18,710 文)	修正学習後	98.67%	97.99%	98.33%	96.00%	95.34%	95.67%
JUMAN		98.88%	98.52%	98.70%	93.93%	93.58%	93.75%
JUMAN+KNP		98.89%	98.53%	98.71%	95.10%	94.75%	94.93%

ステム JUMAN version 3.61[28] に付属の辞書を使った．確率的モデルとして品詞 bigram を使い，二値分類器として 2 次の多項式カーネルの SVM を使った☆．また，JUMAN による解析結果と，それを構文解析システム KNP[26] によって曖昧性を解消したものとの比較を行った．

☆ コーパスからパラメータを自動獲得する確率的モデルに基づいた茶筌とは異なり，JUMAN は人手により設定したパラメータを利用するルールベースのシステムである．そのため，正解の候補に対して必ずしも意味のある順位を得られるわけではないため，本実験では確率的モデルとして JUMAN は使用しなかった．

品詞 bigram に修正学習を適用した結果を**表 13** 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．また訓練データの増加により，修正学習による改善率も上がっていることが分かる．形態素区切りの精度を見ると，修正学習を行った品詞 bigram モデルは JUMAN の結果を下回っている．しかしながら，この京大コーパスの作成には JUMAN を使用し，その際見つかった解析誤りを分析することで JUMAN の改良を行っているため[27]，JUMAN の京大コーパスに対する解析精度は open なデータに対する評価というよりも closed なデータに

表 14 修正学習の副作用
Table 14 Side effects of revision learning.

コーパス	WSJ コーパス	RWCP コーパス		京大コーパス
確率的モデル	T3	品詞 bigram	茶筌	品詞 bigram
正しく修正されたトークン	2,927	2,572	684	909
誤って修正されたトークン	1,795	229	139	66
修正できなかったトークン	6,793	1,515	1,483	1,186

表 15 副作用の例
Table 15 Example of side effects.

テストデータ	
正解	Esso(NNP) said(VBD) the(DT) Whiting(NNP) field(NN) ...
確率的モデルの出力	Esso(NNP) said(VBD) the(DT) Whiting(VBG) field(NN) ...
修正学習後の出力	Esso(NN) said(VBD) the(DT) Whiting(VBG) field(NN) ...
訓練データ	
(文 1) (ノイズ)	Esso(NN) said(VBD) the(DT) fields(NNS) were(VBD) ...
(文 2)	... Group(NNP) ;(:) Esso(NNP) Resources(NNP) Canada(NNP) ...
(文 3)	Three(CD) companies(NNS) ,(,) Esso(NNP) Resources(NNPS) ...

対する評価に近いと思われる．また，JUMAN では連語の登録等により解析精度を高める工夫も行われている[24] こと等も考慮すると，特別なチューニングは行っていない修正学習後の形態素区切り精度は低いとはいえないだろう．

4.3.4 修正学習の副作用

これまでに説明した 3 つのコーパスに対する実験では，いずれの場合も修正学習により全体的な精度は改善された．しかしながら，なかには確率的モデルが正しい品詞を認識したにもかかわらず，SVM により誤った品詞に修正してしまうケースも存在すると考えられる．そこで，これまで行った実験について，(1) 確率的モデルが誤った品詞を認識したが修正学習により正しい品詞に修正したもの，(2) 確率的モデルは正しい品詞を認識したが修正学習により誤った品詞に修正したもの，(3) 確率的モデルが誤った品詞を認識したが修正学習でも正しく修正できず誤ったままのもの，という 3 つの場合の数をそれぞれ調べた．結果を**表 14** に示す．

この結果を見ると，使用したコーパスや確率的モデルによって差はあるが，修正学習を行ったことによる副作用が少なくないことが分かる．確率的モデルとして品詞 bigram を用いた場合には副作用の割合が小さいが，これは品詞 bigram による誤りが多いために，SVM によって正しく修正される割合が高くなっているためと考えられる．

修正学習法による副作用が起こる原因の 1 つとして，学習モデルのノイズに対する頑健性があげられる．つまり，SVM のような表現力の高いモデルは，より単純な n-gram 等のモデルに比べて訓練データ中のノイズに弱くなるという問題がある．**表 15** に WSJ コーパスを用いた実験で副作用の起きた例と，その原因と思われる訓練データ中の文を示す（括弧内は Penn Treebank の品詞を表す）．この例では，テストデータ中の “Esso” という固有名詞 (NNP) に対して，確率的モデル（品詞 trigram）では正しい品詞を認識できたにもかかわらず修正学習により普通名詞 (NN) と誤った品詞に修正している．このとき訓練データ中で “Esso” という単語を含む文は，表 15 の (文 1)，(文 2)，(文 3) の 3 つだけが存在した．ここで，(文 1) では “Esso” の品詞が普通名詞 (NN) になっているが，これはコーパスを作成した際の誤りと思われる．確率的モデルとして使用した品詞 trigram では，単語の出現確率と品詞の推移確率のみを使用して最も尤度の高い解を求めるため，学習データ中に偶然出現したノイズの影響は比較的受けにくい．実際に，表 15 の例でも，ノイズの影響を受けることなく正しい解を認識している．一方で，SVM は n-gram モデルとは異なり，訓練データ中での事象の出現頻度にかかわらず，テストデータと最も類似した訓練データ中の事例のクラスを解として優先する性質がある．この実験では，注目している単語の前後 2 つの単語と品詞を素性としたが，テストデータと (文 1) における単語 “Esso” の前後 2 つの文脈は同一であることが分かる（前 2 つの文脈は，どちらも文頭であるため同一であると見なせる）．このため，SVM を用いて修正学習を行ったことで，本来誤りである (文 1) の品詞 (NN) を解として優先したと考えられる．

このように，訓練データ中に少数しか出現しない例外的な現象でも学習できるという性質は本来は SVM の利点であるが，一方で訓練データ中のノイズの影響を受けやすくし，副作用を引き起こす原因となりうることが分かる．

4.4 関連研究

ここでは，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 の変換規則に基づく誤り駆動方式がある[6]．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っているという点が大きく異なる．また，本手法では初めに確率的モデルによって順番を付け，二値分類器によってその結果の正誤を判定するという 2 段階の方式であるのに対し，Brill の方法では書き換え規則を何回も適用するものになっている．

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たりしようとする試みは近年さかんに行われている[3]．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．1 つは並列的な方法（multiexpert method）であり，もう 1 つは直列的な方法（multistage method）である．前者は，複数の学習器でそれぞれ独立に学習，評価を行い，すべ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われており[16),22]，Brill ら[7] は最大エントロピー法，変換規則に基づく誤り駆動方式，unigram，trigram の 4 つのモデルに基づいた品詞タグ付けシステムを混合することにより，それらの個々のシステムよりも高い精度（WSJ コーパスに対して 97.2%）を得たと報告している．後者のアプローチに関しては，カスケーディング[2] 等の方法が提案されており，Even-Zohar ら[12] はシーケンシャル学習モデルを提案し品詞タグ付けに適用している．この方法と修正学習法を比較した場合，彼らの方法では 1 つ以上の学習器を同じ方法で適用していくのに対し，修正学習法では確率的モデルが順番を付け二値分類器がそれを修正するというように役割が分かれている点が大きく異なっている．さらに彼らの方法では，初めの段階にある学習器が誤った場合は，そのまま誤った出力を行うが，修正学習法の場合は 2 段目の二値分類器によってそれを訂正できる可能性がある．

並列的なアプローチでは各学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すべての訓練データで学習を行い，すべてのテストデータで評価を行うため，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．

SVM 自体の学習や評価に要する計算量を削減する方法に関してはこれまで様々な手法が試みられている[8),9),11),13),14),17]．本手法は学習器自体ではなく学習器の適用法に焦点を当てたものであり，学習時と評価時の両方の計算量を削減することができる．また，既存の SVM 自体を高速化する手法と組み合わせて使用したり，SVM 以外の二値分類器を用いたりすることも可能である．

5. まとめ

本稿では，SVM を利用した形態素解析を試みた．そして，One-versus-Rest 法によって英語の未知語の品詞推定と英語の品詞タグ付けを行った結果，従来手法を上回る高い精度を得た．しかし一方で，膨大な計算量を要することや，日本語の形態素解析に適用できない等の問題も明らかになった．そこでこれらを解決するために，計算量の小さいモデルと表現力の高いモデルを組み合わせて利用し，One-versus-Rest 法とは異なった方法で多値分類問題を解く修正学習法を提案し，英語の品詞タグ付けと日本語の形態素解析に適用した結果，計算量を抑えつつ高い精度が得られた．特に，形態素解析は多数の品詞（RWCP コーパスの場合 88 個）を扱う必要があるために大規模な多値分類問題となるが，このような問題に対しても本手法は有効であることが確認できた．複数のコーパスを用いた実験を通じて，SVM を用いて単語等の細かい素性も考慮した学習を行うことで，人手による細かいチューニングを必要とせずに高い精度で形態素解析を行えることを確認した．

本稿で提案した修正学習法は，品詞 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，形態素解析以外の用途にも用いることができる．ただし，修正学習法による精度は One-versus-Rest 法によるものより低かったため，この差を小さくすることは今後

の課題である．また，英語や日本語の形態素解析はかなり高い精度を達成するまでになってきているが，英語品詞タグ付けに関してはいえば未知語に対する精度は既知語に比べてまだ十分でなく，日本語形態素解析における未知語処理は分かち書きの難しさから研究の余地が多く残されている．さらにコーパス中に存在する誤りも形態素解析の失敗の大きな原因となっているため，これらの問題への対処も今後の重要な課題といえる．

謝辞 ICOPOST を作成し公開されている Ingo Schröder 氏と，形態素解析システム茶筌について数々のアドバイスをいただいた奈良先端科学技術大学院大学情報科学研究科の浅原正幸氏に深く感謝いたします．

参 考 文 献

1) Allwein, E.L., Schapire, R.E. and Singer, Y.: Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers, *Proc. 17th International Conference on Machine Learning*, pp.9–16 (2000).
2) Alpaydin, E. and Kaynak, C.: Cascading Classifiers, *Kybernetika*, Vol.34, No.4, pp.369–374 (1998).
3) Alpaydm, E.: Techniques for Combining Multiple Learners, *Proc. Engineering of Intelligent Systems '98 Conference* (1998).
4) Asahara, M. and Matsumoto, Y.: Extended Models and Tools for High-performance Part-of-Speech Tagger, *Proc. 18th International Conference on Computational Linguistics*, pp.21–27 (2000).
5) Brants, T.: TnT—A Statistical Part-of-Speech Tagger, *Proc. 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association of Computational Linguistics*, pp.224–231 (2000).
6) 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).
7) Brill, E. and Wu, J.: Classifier Combination for Improved Lexical Disambiguation, *Proc. 36th Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pp.191–195 (1998).
8) Burges, C.J.C.: Simplified Support Vector Decision Rules, *Proc. 13th International Conference on Machine Learning*, pp.71–77 (1996).
9) Burges, C.J.C. and Schölkopf, B.: Improving the Accuracy and Speed of Support Vector Machines, *Advances in Neural Information Processing Systems*, Vol.9, The MIT Press (1997).
10) Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M.: Equations for Part-of-Speech Tagging, *Proc. 11th National Conference on Artificial Intelligence*, pp.784–789 (1993).
11) Downs, T., Gates, K.E. and Masters, A.: Exact Simplification of Support Vector Solutions, *Journal of Machine Learning Research*, Vol.2, pp.293–297 (2001).
12) Even-Zohar, Y. and Roth, D.: A Sequential Model for Multi-Class Classification, *Proc. 2001 Conference on Empirical Methods in Natural Language Processing*, pp.10–19 (2001).
13) Isozaki, H. and Kazawa, H.: Efficient Support Vector Classifiers for Named Entity Recognition, *Proc. 19th International Conference on Computational Linguistics*, pp.390–396 (2002).
14) Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical, *Advances in Kernel Methods — Support Vector Learning*, MIT Press (1998).
15) Kudoh, T. and Matsumoto, Y.: Use of Support Vector Learning for Chunk Identification, *Proc. 4th Conference on Computational Natural Language Learning*, pp.142–144 (2000).
16) Màrquez, L., Rodríguez, H., Carmona, J. and Montolio, J.: Improving POS Tagging Using Machine-Learning Techniques, *Proc. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp.53–62 (1999).
17) Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization, *Advances in Kernel Methods—Support Vector Learning*, MIT Press (1998).
18) Ratnaparkhi, A.: A Maximum Entropy Model for Part-of-Speech Tagging, *Proc. Conference on Empirical Methods in Natural Language Processing*, pp.133–142 (1996).
19) Roth, D. and Zelenko, D.: Part of Speech Tagging Using a Network of Linear Separators, *Proc. joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pp.1136–1142 (1998).
20) Schmid, H.: Probabilistic Part-of-Speech Tagging Using Decision Trees, *Proc. International Conference on New Methods in Language Processing*, pp.44–49 (1994).
21) Schröder, I.: ICOPOST—Ingo's Collection Of

POS Taggers (2001). http://nats-www.informatik.uni-hamburg.de/~ingo/icopost/
22) 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).
23) Vapnik, V.: *Statistical Learning Theory*, Springer (1998).
24) 山地 治，黒橋禎夫，長尾 眞：連語登録による形態素解析システム JUMAN の精度向上，言語処理学会第 2 回年次大会，pp.73–76 (1996).
25) 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).
26) 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).
27) 黒橋禎夫，長尾 眞：京都大学テキストコーパス・プロジェクト，言語処理学会第 3 回年次大会，pp.115–118 (1997).
28) 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).
29) 平博 順，春野雅彦：Support Vector Machine によるテキスト分類における属性選択，情報処理学会論文誌，Vol.41, No.4, pp.1113–1123 (2000).
30) 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).
31) 松本裕治，浅原正幸：IPADIC ユーザーズマニュアル version 2.2.4，奈良先端科学技術大学院大学松本研究室 (2001).

(平成 14 年 9 月 11 日受付)
(平成 15 年 3 月 4 日採録)

中川 哲治（正会員）

1978 年生．2000 年筑波大学第三学群情報学類卒業．2002 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年沖電気工業株式会社に入社．統計的自然言語処理および機械学習に興味を持つ．

工藤 拓（学生会員）

1976 年生．1999 年京都大学工学部電気電子工学科卒業．2001 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，同年同大学院博士後期課程に進学．2001 年度本学会山下記念研究賞受賞．統計的自然言語処理，テキストマイニング，機械学習に興味を持つ．

松本 裕治（正会員）

1955 年生．1977 年京都大学工学部情報工学科卒業．1979 年同大学大学院工学研究科修士課程情報工学専攻修了．同年電子技術総合研究所入所．1984 年〜1985 年英国インペリアルカレッジ客員研究員．1985 年〜1987 年（財）新世代コンピュータ技術開発機構に出向．京都大学助教授を経て，1993 年より奈良先端科学技術大学院大学教授，現在に至る．工学博士．専門は自然言語処理．人工知能学会，日本ソフトウェア科学会，言語処理学会，認知科学会，AAAI，ACL，ACM 各会員．