決定リストを弱学習器としたアダブーストによる日本語単語分割

新納浩幸†

本論文では決定リストを弱学習器としたアダブーストによる日本語単語分割法を提案する．日本語単語分割は，入力文の各文字の間に単語区切りを置くか置かないかの問題とみなすことで，分類問題として定式化できる．この分類問題を決定リストを利用して解くことで単語分割が行える．ここでは決定リストで利用する属性に辞書情報を含めない．そのためここでの単語分割は未知語の問題を受けないという長所がある．更に単語分割を分類問題として解く場合，近年研究の盛んなアダブーストの手法を適用できる．アダブーストを用いることで，決定リストの精度を高めることができる．実験では，京大コーパス(約4万文) を利用して決定リストを作成した．この決定リストによる単語分割の正解率は 97.52% であった．この値は、同じ訓練データから構築した tri-gram モデルに基づく単語分割法での正解率 92.76% を大きく上回った．またアダブーストを利用することで精度が 98.49% にまで向上させることができた．また作成した単語分割システムは未知語の検出能力が高いことも確認できた．

キーワード： 単語分割，分類問題，決定リスト，アダブースト

Japanese word segmentation by Adaboost using the decision list as the weak learner

Hiroyuki Shinnou†

In this paper, we propose the new method of Japanese word segmentation by Adaboost using the decision list as the weak learner. The word segmentation is regarded as the classification problem of judging whether the word boundary exists between two characters or not. By solving the problem by the decision list method, we can conduct Japanese word segmentation. Our method has the advantage not to suffer the unknown word problem because we do not use dictionary information as an attribute of our decision list. Moreover, by taking this approach we can use Adaboost which is actively researched in the machine learning domain recently. Adaboost improves the precision of our decision list. In experiments, we built the decision list through Kyoto University Corpus (about 40K sentences). The precision of this decision list was 97.52%. This values was much higher than the precision of character based tri-gram model, 92.76%. By using Adaboost method, our precision was improved to 98.49%. Furthermore, our word segmentation system was excellent in detecting unknown words.

KeyWords: *Word segmentation, classification problem, decision list, Adaboost*

† 茨城大学工学部システム工学科, Faculty of Engineering, Ibaraki University Department of Systems Engineering

1　はじめに

本論文では日本語単語分割を分類問題とみなし，決定リストを利用してその問題を解く．このアプローチは文字ベースの手法の一種となり，未知語の問題を受けないという長所がある．また分類問題ととらえることで，ブースティングの手法が適用できる．その結果，単独の決定リストを利用するよりも，さらに精度を向上させることができる．

日本語形態素解析は，日本語情報処理において必須の要素技術であり，その重要性は明らかである．日本語形態素解析は単語分割と分割された単語への品詞付与という2 つのタスクをもつ．正しい単語分割からは英語の品詞タガーなどの技術を利用して，高精度に品詞付与ができるために，日本語形態素解析の本質的に困難な部分は単語分割である．特に未知語の問題が深刻である．未知語の問題とは，辞書に登録されていない単語の出現によりその単語とその単語の前後での単語分割が誤るという問題である．

未知語の問題に対処する一つの方法として，文字ベースの単語分割手法がある．文字ベースの手法とは，辞書を使わずに，各文字間に単語境界が存在するかどうかを判定することで単語分割を行う手法である．従来，文字ベースの手法としては，文字ベースの HMM (Hidden Markov Model)が提案されている．文字ベースの HMM は，状態として文字間に単語境界が存在する（ 状態1 ）としない（ 状態0 ）の2 つを設定し，状態間を遷移するときに各文字が出力されるモデルである．単語分割は遷移した状態列を推定することで行える．文字ベースの HMM では状態 a から状態 b に移るときに文字 c を出力する確率を訓練データから得る．本質的にこの確率の精度が単語分割の精度を左右する．通常その確率を計算するために tri-gram モデルを利用するが，常識的に考えても，前2 文字から次の文字を予測することは難しく，文字ベースの HMM 単独ではそれほどの精度は期待できない．このため，様々な工夫を付加する必要がある (山本 増山 1997; Tsuji and Kageura 1997; 小田 北 1998)．

本論文では単語分割を HMM によりモデル化して解くのではなく，分類問題として定式化して解く．先ほども述べたように，日本語単語分割は，各文字間に単語境界が存在する（ クラス +1）か存在しない（ クラス −1）かを判定する問題であり，これは分類問題に他ならない．分類問題を解くために設定する属性として，辞書情報を使わないことで，文字ベースの単語分割手法と同様未知語の問題を受けない．また分類問題として見なすことで， n-gram モデルでは利用の困難であった様々な属性を判定の材料として利用可能になる．さらに，分類問題は機械学習や統計学で活発に研究されている問題であり，それらの研究成果を直接利用することができる．

本論文では単語分割を分類問題と見なし，分類問題に対する帰納学習手法の一つである決定リスト (Yarowsky 1994)を用いて，その問題を解く．さらに，近年，機械学習の研究分野では弱学習器を組み合わせて強学習器をつくるブースティングの研究が盛んである．ここではその代表的な手法であるアダブースト (Freund and Schapire 1997)を本問題に対して適用する．

実験では，タグつきのコーパスである京大コーパス（ 約4 万文）を訓練データとして，決

定リストを作成した．その決定リストを利用した単語分割は，同じデータから学習させた文字 tri-gram モデルに基づく単語分割法（ 文字ベースの HMM の一種）よりも高い精度を示した．さらに，アダブーストを利用することで，単独の決定リストよりも高い精度を得ることができた．また本手法の未知語の検出率が高いことも確認した．

2　決定リストによる単語分割

2.1　単語分割と分類問題

n 文字からなる入力文を $s = c_1 c_2 \cdots c_n$（ 各 c_i は文字を表す）とすると，日本語単語分割は文字 c_i と c_{i+1} の間（ b_i と名付ける）に単語境界がある $(+1)$ かない (-1) かを与えることによって行える。つまり b_i（ $i = 1, 2, \cdots, n-1$ ）に $+1$ か -1 を与える分類問題としてとらえられる．例えば,「太郎は海でアイスクリームを食べた。」という文に対しては，図 1 のように各文字間にクラス $+1$ あるいは -1 を付与し，$+1$ の部分を単語境界に置き換えることにより単語分割が行える．

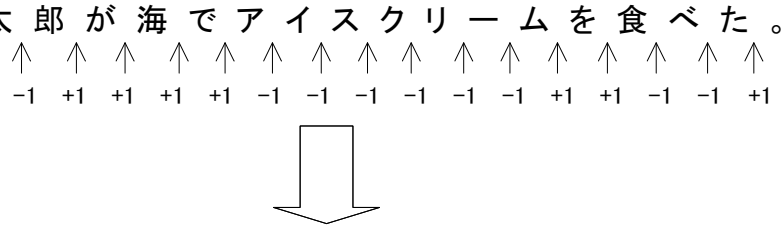

図 1　クラスの付与による単語分割
図 1　Word segmentation by class assignment

分類問題を解く手法は様々なものがある．どの手法が優れているかは問題に依存するために一概には言えない．本論文では決定リストを利用して上記の分類問題を解く．

2.2　決定リストの構築

決定リストは帰納学習手法の一種であり，正解付きの訓練データから，分類規則を学習する．決定リストの場合，分類規則は証拠とクラスの組の順序付きの表となる．ここで証拠とは属性とその属性の値の組である．実際の分類はリストの上位のものから順に，その証拠があるかどうかを調べ，その証拠があれば，それに対応するクラスを出力する．

決定リストの作成は概ね以下の手順による．

step 1　属性を設定する．

例えばn個の属性を$att_1, att_2, \cdots, att_n$とする。

step 2　訓練データから証拠とクラスの組の頻度を調べる．

訓練データ中のあるデータの属性attの値がaであるとし，そのデータのクラスがCだとする．その場合，(att, a)という証拠とクラスCの組$((att, a), C)$の頻度に1を足す．これを訓練データ中の全データに対する全属性について行う．

step 3　証拠の判別力と分類クラスを導く．

$((att, a), C)$の頻度がf_Cであった場合，f_Cの最大値を与える$\hat{C}$が証拠(att, a)に対する分類クラスとなる．またそのときの判別力$pw(att, a)$は以下で定義される．

$$pw((att, a)) = \log \frac{f_{\hat{C}}}{\sum_{C \neq \hat{C}} f_C}$$

step 4　判別力の順に並べる．

全ての証拠と分類クラスの組を判別力の大きい順に並べる．これによって作成できた表が決定リストである．

2.3　属性の設定

各文字間b_iがどのクラスに属するかを判断する材料が属性である．本論文ではb_iの属性として，表1の7種類を用意した．

表 1　設定した属性
表 1　Setting attributes

属性	値	
att_1	文字列	$c_{i-1}c_ic_{i+1}$
att_2	文字列	$c_ic_{i+1}c_{i+2}$
att_3	文字列	$c_{i-1}c_i$
att_4	文字列	c_ic_{i+1}
att_5	文字列	$c_{i+1}c_{i+2}$
att_6	字種の接続関係1	$((c_i$の大分類字種$), (c_{i+1}$の大分類字種$))$
att_7	字種の接続関係2	$((c_i$の細分類字種$), (c_{i+1}$の細分類字種$))$

6，7番目の属性として，字種の情報を利用している形になっている．ここでは字種を大分類と細分類の二つの観点から分類した．字種の大分類は6番目の属性，字種の細分類は7番目の属性で利用した．

字種の大分類は表2に示した9種類である．

表 2　大分類字種
表 2　Classification of character types

字種	意味	例
平	平仮名	あ，い，う，　…
カ	カタカナ	ア，イ，ウ，　…
数	漢数字	一，二，…，百，千，…
漢	漢字	亜，位，卯，　…
N	英数字	０，１，２，…
ア	アルファベット	Ａ，Ｂ，Ｃ，…
記	記号	、，。，「，　…
○	小丸かゼロ	○
○	大丸かゼロ	○

字種の細分類は大分類の平仮名の部分をその文字自身にしたものである．

また注意として，本論文の決定リストでは $default$ の証拠を導入していない．決定リストでは通常 $default$ という証拠を設けて，それ以下の判別力の証拠は表には入れない．$default$ は文脈上の証拠が決定リストに存在しない場合の処理ととらえられるが，ここでは大分類の字種の情報が必ずヒットするので，$default$ の証拠を含める必要がない．6 番目の属性からの証拠の最下位のものが，決定リストの最下位の証拠となる．

2.4　利用例

決定リストの利用例を示す．例えば「太郎は海でアイスクリームを食べた。」という入力文の 5 番目の文字 “で” と 6 番目の文字 “ア” の間，つまり b_5 にクラス $+1$ あるいは -1 を与えてみる．b_5 の持つ証拠は以下の 7 種である．

$$(att_1,\text{”海でア”}),\ (att_2,\text{”でアイ”}),\ (att_3,\text{”海で”}),$$
$$(att_4,\text{”でア”}),\ (att_5,\text{”アイ”}),\ (att_6,\text{”平カ”}),\ (att_7,\text{”でカ”})$$

後述する実験で得られた決定リストを用いると，各証拠の分類クラスと判別力は以下の通りである．

表 3　クラス判別の例
表 3　Example of class judgement

証拠	分類クラス	判別力
$(att_1,$"海でア")	–	–
$(att_2,$"でアイ")	–	–
$(att_3,$"海で")	+1	2.74377
$(att_4,$"でア")	+1	5.83188
$(att_5,$"アイ")	+1	1.64565
$(att_6,$"平カ")	+1	6.33293
$(att_7,$"でカ")	+1	8.64488

表の中で “–” の記号のものは，決定リスト中にその証拠がないことをあらわす．また本来ならば，決定リスト中の順位を求めなければならないが，ここでは相対的な順位関係だけが必要であり，順位の値自体は必要でない．判別力の最も大きなものが最上位の順位になるはずである．この場合，証拠 $(att_7,$"でカ") が最も大きな判別力を持つので，この証拠の分類クラス +1 が判定結果となる．つまり b_5 には単語境界を置くと判定する．

3　アダブーストの利用

精度の低い分類規則を組み合わせて精度の高い分類規則を得る方式をブースティングという．アダブーストはブースティング方式の一つであり，現在まで多くの理論的検証と実験的実証から有効性が示されている．

アダブーストのアルゴリズムを図 2 に示す．分類クラス (図 2 の Y) をここでは $\{+1, -1\}$ の 2 値とする．また訓練データを $(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)$ で表す．ここで各 x_i はデータを表し，y_i はデータ x_i のクラスである．具体的に y_i は $+1$ あるいは -1 の値である．この訓練データに対して，分類問題に対する学習アルゴリズム，例えば，決定木や決定リストなどを適用して，分類規則 h_1 を学習する．得られた分類規則 h_1 を訓練データに適用すると，h_1 によって各 x_i の判定クラスが得られる．今，x_i の実際のクラス y_i は与えられているので，分類規則 h_1 が各 x_i に対して正しい判定を行ったかどうかを調べられる．これによって不正解のデータを集め，それら不正解のデータに対してある重みを付加して，訓練データ $(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)$ を再構成する．そしてこの再構成された訓練データに対して，再び学習アルゴリズムを適用して，分類規則 h_2 を学習する．これを T 回繰り返す．この繰り返しによって，T 組の分類規則 $h_1, h_2, \cdots, h_T$ が得られる．実際の判定は入力データに対して各分類規則が出力するクラスの重み付き多数決により行われる．

例えば，$T = 3$ とし，入力データ x に対して，分類器 h_1 による判定クラスが $+1$，h_2 によ

る判定クラスが -1，h_3 による判定クラスが $+1$ であり，各重みが 1 ， 2.0 ，2.2 であった場合，重み付き多数決の結果は $+1.2$ である．最終的な判定クラスは総和の符合により求まる．この例の場合，符合は正であるので，$+1$ が判定クラスになる．

アダブーストのポイントは不正解のデータに課す重みの与え方である．概略，得られた分類規則の誤り確率(図2における ϵ_t) が小さいほど重みが大きくなるように設定している．

Given: $(x_1, y_1), \cdots, (x_m, y_m)$ where $x_i \in X$,$y_i \in Y = \{1, -1\}$
Initialize $Di(i) = 1/m$
For $t = 1, \cdots, T$

- Train weak learner using distribution D_t
- Get weak hypothesis $h_t : X \to Y$ with error

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

- Choose $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if \quad h_t(x_i) = y_i \\ e^{\alpha_t} & if \quad h_t(x_i) \neq y_i \end{cases}$$

  where Z_t is a normalization factor

Output the final hypothesis:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

図 2 アダブースト
図 2 AdaBoost

本論文では．分類問題に対する学習アルゴリズムを決定リストに設定する．不正解データに与える重みをどのように反映させるかが問題である．ここでは，重みを頻度として与えることにした．例えば,「 太郎が東京へ行く。」という文に以下のように単語境界 “/” が置かれたものが訓練データである．

太郎/が/東京/へ/行く /。

今，4 番目の文字 “東” と 5 番目の文字 “京” の間，つまり b_4 に対する証拠は以下の通りである．

$$(att_1,\text{”が東京”}),\ (att_2,\text{”東京へ”}),\ (att_3,\text{”が東”}),$$
$$(att_4,\text{”東京”}),\ (att_5,\text{”京へ”}),\ (att_6,\text{”漢漢”}),\ (att_7,\text{”漢漢”})$$

“東” と “京” の間には，単語境界がないので，クラスは -1である．そして，決定リスト作成の step 2 で示したように，以下の証拠の頻度に 1 が足される．

$$((att_1,\text{”が東京”}),-1),\ ((att_2,\text{”東京へ”}),-1),\ ((att_3,\text{”が東”}),-1),$$
$$((att_4,\text{”東京”}),-1),\ ((att_5,\text{”京へ”}),-1),\ ((att_6,\text{”漢漢”}),-1),\ ((att_7,\text{”漢漢”}),-1)$$

この頻度に加算される 1 という数値に重みを反映させる．

例えば，決定リスト h_k により上記例文の 4 番目の文字 “東” と 5 番目の文字 “京” の間の判定クラスが $+1$と判定された場合，この判定は不正解である．そこで次の決定リスト h_{k+1}を作成するときに，上記の7 つの各証拠の頻度に 1 ではなく，重み自身を加える．

つまり決定リストを作成する際には各訓練データには重みがついているとして，その重みが決定リスト作成の step 2 で各証拠と正解の組に付加する数値とする．図2のアルゴリズムでは正規化するために重みの総和が1 になっているが，ここでは重みの最小値が1 となるようにして計算を簡単にした．このため最初の決定リストを作成する際の各訓練データの重みは1 であり，2 回目では正解のデータの重みは1 で変化せず，不正解の部分の重みが大きくなる．

4　実験

4.1　文字 n-gram モデルに基づく単語分割法との比較

ここでは決定リストを利用した単語分割の有効性を示すために，文字 n-gram モデルに基づく単語分割法(小田・ 北 1998)との比較を行う．文字 n-gram モデルに基づく単語分割法では，概略，単語境界を付与した訓練データにおいて，単語境界の記号自体も一つの特殊文字として考えて，ある文字列の後に単語境界が生じる確率あるいは生じない確率を文字 n-gram モデルに基づいて計算する．最終的には Viterbi アルゴリズムなどの動的計画法を利用して，文字列の出現確率が最大になるように単語境界のあるなしの列を決定する．これは文字ベースの HMM において，遷移確率やシンボル出力確率をある確率モデルに基づいて計算したものと同等である．

訓練データとしては京大コーパス (約 4 万文)を利用した．京大コーパスは人手でタグをつけた

コーパスであり，正解付きの訓練データとして利用できる．京大コーパスの中から 950117.KNP というファイルに納められた 1,234 文[1]をテストデータとした．結果，訓練データは京大コーパスからテストデータを除いた 35,717 文である．テストデータ 1,234 文の中には，単語境界を置くか置かないかを判定する位置が 56,411 個所存在する．この 56,411 個所に対して正しいクラスを付与できた割合を正解率とする．

訓練データから文字 tri-gram 確率を求めるために CMU-Cambridge Toolkit [2]を利用した．スムージングの手法としては Witten-Bell discounting を用い，カットオフは頻度 0 と設定した (北 1999)．

文字 tri-gram 確率から tri-gram モデルに基づく単語分割法を実装したシステムを作成し，テストデータに対して単語分割を行った．結果，56,411 個所の判定位置について，52,328 個所で正しい判定を行い，4,083 個所で誤った判定を行った．つまり正解率は 92.76% であった．

次に上記の訓練データを利用して本論文で提案した決定リストを作成した．頻度 7 以下の証拠は間引いた．作成できた決定リストの大きさは 136,114 であった．この決定リストによりテストデータに対して単語分割を行った．結果として，56,411 個所の判定位置について，55,015 個所で正しい判定を行い，1,396 個所で誤った判定を行った．つまり正解率は 97.52% であった．この値は tri-gram モデルに基づく単語分割法の正解率 92.76% を大きく上回っており，本手法の有効性が示せた．

4.2 ブースティングの効果

前述したアダブーストにより，決定リストのブースティングを行った．ブースティングの回数を横軸に，テストデータに対する正解率 % を縦軸にしたグラフが 図3 である．

図3 からわかるように，ブースティングにより 3 組の決定リストを作成し，それらの重み付き多数決によって判別した結果が最も優れていた．そのとき 56,411 個所の判定位置について，55,560 個所で正しい判定を行い，851 個所で誤った判定を行った．つまり正解率は 98.49% まで向上した．

4.3 未知語の検出

ブースティングにより 3 組の決定リストを作成し，それらの重み付き多数決によって各文字間に単語境界の有無を判定する手法（以下これを本手法と呼ぶ）が，本実験において，どの程度の未知語を検出できたか調べる．

前述した訓練データ 35,717 文とテストデータ 1,234 文の正確な単語分割結果から，それぞれに含まれている単語文字列を取り出した．ここでいう単語文字列とは，単純に単語分割され

1 ここではコーパス中の記号 EOS の数を文の数としている．句点 “。” の数ではないことを注意しておく．
2 CMU-Cambridge Toolkit は以下のアドレスから入手可能．http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

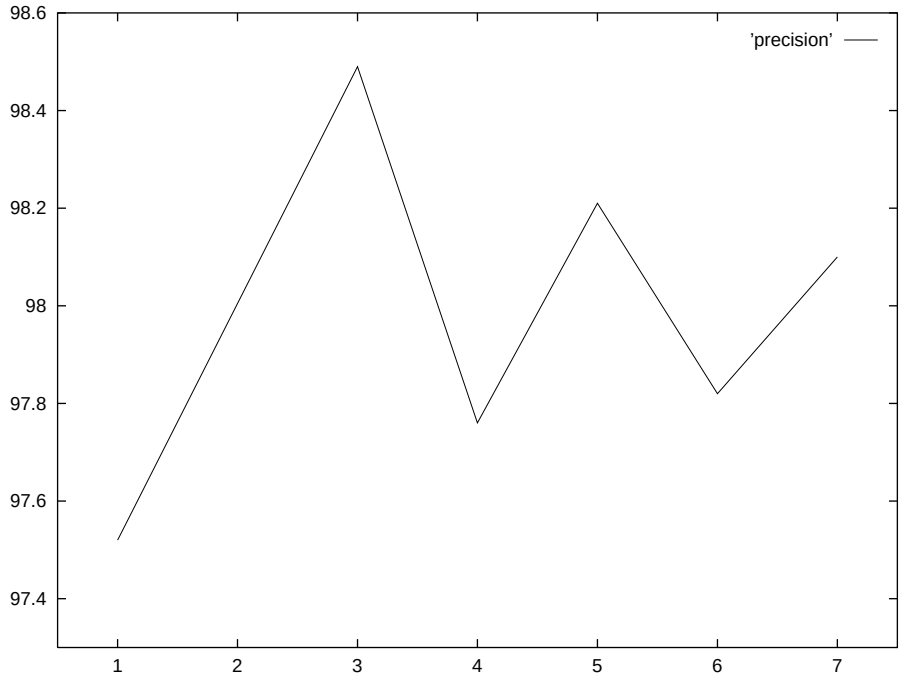

図 3　ブースティングによる正解率
図 3　Precision by boosting

た分割要素の文字列のことである．つまり用言の活用語尾が異なるものも，異なる単語文字列として取り出すことに注意する．結果，訓練データには 914,392 個（ 41,890 種類）の単語文字列，テストデータには 32,764 個（ 6,479 種類）の単語文字列が存在した．そしてテストデータには含まれるが，訓練データには含まれない単語文字列が 1,024 個（ 832 種類）存在した．この 1,024 個（ 832 種類）の単語文字列が本実験における未知語となる．

結論から述べると，本手法によりこの 1,024 個（ 832 種類）の未知語の中で，正しく検出できたものは 688 個（ 562 種類），つまり個数で 67.2%，種類数で 67.5% の検出率であった．

検出できた未知語の中には，字種区切りのような単純なヒューリスティクスから検出できるものも存在するので，本手法の未知語検出が，実質どの程度の有用性があるのかを示すために，対象の未知語を以下のように 9 タイプに分類した．

(1) 用言であり，その原型を同じとする単語が訓練データに含まれる（ 124 個（ 123 種類））．

例えば,「 押しつぶした」という単語文字列は，テストデータには含まれるが，訓練データには含まれないために，本手法では未知語として扱われる．しかし通常の辞書を利用したシステムでは,「 押しつぶした」の原型「 押しつぶす」が辞書に登録されていれば，正しく解析できる．訓練データには,「 押しつぶす」の語尾変化形である単語文字列「 押

しつぶして」が含まれている．そこで，通常のシステムの辞書には，原型「押しつぶす」が登録されていたと考え，「押しつぶした」は正しく解析できると考える．

ここでは，このようなタイプの未知語は，通常のシステムの用言の語尾変化の規則によって検出できるタイプの未知語として考える．

(2) 用言であり，その原型を同じとする単語が訓練データに含まれない（94個（91種類））．
例えば，「飲みすぎて」という単語文字列は，テストデータには含まれるが，訓練データには含まれない．しかも (1) の場合とは異なり，「飲みすぎて」の原型「飲みすぎる」を語尾変化させた単語文字列も訓練データに含まれない．これは通常のシステムにおいても未知語となるものである．

(3) 数値表現となっている（44個（41種類））． 例えば，「一万九千百八十五」や「２７・７」という単語文字列は未知語となっているが，通常のシステムはこれらの表現を数値表現として認識できる規則を持っている．この種の未知語も通常のシステムで検出できるタイプの未知語とする．

(4) アルファベットのみで構成される（7個（3種類））． 「ＡＣ」「ＯＥＫ」「ＰＡＨ」の単語文字列である．これらは字種区切りのような単純なヒューリスティクスから通常のシステムでも検出可能である．

(5) カタカナのみで構成される（210個（156種類））． 例えば「アロマセラピスト」や「スーザン」のような単語文字列である．これらも字種区切りのような単純なヒューリスティクスから通常のシステムでも検出可能である．

(6) 平仮名のみで構成される（38個（32種類））． 例えば，「ごあいさつ」や「ぞろぞろ」のような単語文字列である．これらの検出は通常のシステムでは不可能である．

(7) 漢字1文字で構成される（21個（17種類））． 例えば，「魁」や「鋼」のような単語文字列である．通常のシステムでも未知語となるが，単語分割の他の候補が生じないために，結果的に正しく単語分割できる場合も多い．

(8) 漢字のみで構成される（426個（310種類））． 例えば，「重文」や「三井造船」のような単語文字列である．これらの検出は通常のシステムでは不可能である[3]．

(9) 複数の字種から構成される（64個（59種類））． 例えば，「寝泊まり」や「亡き後」のような単語文字列である．これらの検出は通常のシステムでは不可能である．

上記 9 タイプの未知語の本手法による検出結果を表 4に示す．同時に通常のシステムで想定できる検出結果も示す．

3 例えば，漢字1 文字からなる未知語と既知語を全体として未知語として認識できる可能性が指摘された．しかしそのヒューリスティクスがどの程度妥当かは疑問がある．また，その場合 (7) との区別がつかない．ここでは多少強引だが，(8) は既存のシステムでは検出不可能とした．

表 4　未知語の検出
表 4　Detection of unknown words

タイプ	総出現数	本手法による検出	通常のシステムによる検出
(1) 辞書登録の用言	124	101	124
(2) 辞書未登録の用言	94	57	0
(3) 数値表現	44	40	44
(4) アルファベット列	7	5	7
(5) カタカナ列	210	188	210
(6) 平仮名列	38	19	0
(7) 漢字1 文字	21	4	21
(8) 漢字列	426	246	0
(9) 複数の字種	64	28	0
合計	1,024	688 (67.2%)	406 (39.6%)

表 4に示すように通常のシステムの検出率は 39.6% であり，本システムの検出率 67.2% と大きく差がある．これは本システムの未知語の検出能力の高さを示している．また通常のシステムにより検出できるとした未知語のタイプ (1),(3),(4),(5),(7) に対して，本手法の検出率は 83.3% であり，通常のシステムにより検出できる未知語の多くは本システムでも検出できると考えられる．

5　考察

本手法での判別の出力は 2 値であり，判別に使った判別力の値自体は利用されていない．テストデータに対して判別力の値による正解率を調べるために，以下の調査を行った．テストデータには 56,411 個所の判定位置があるが，0 以上 1 未満の間の判別力で判定された位置は 83 個所であり，その正解率は 57.83% であった．同様にして，1 以上 2 未満の間，2 以上 3 未満の間という具合いに順に調べていった結果を示したものが図 4である．このグラフからもわかるように，判別に利用した判別力が小さいほど誤る確率が高くなる．このような判別力の値を利用して，さらに誤りを減らせる工夫も可能であろう．

また本論文では分類問題の解法として決定リストを利用したが，他の手法，例えば，決定木 (Quinlan 1993) や最大エントロピー法 (Ratnaparkhi 1998) の利用も可能である．ただし本論文で利用した属性にあたるものを，それらの手法では単純には利用できない．決定木を利用する場合，属性の数は 7種類であり問題ないが，bi-gram あるいは tri-gram にあたる属性の値の種類数が非常に多い．このため決定木の各ノードから出る枝の数が膨大になり，現実的には決定木を作成できない．また最大エントロピー法では素性の設定と素性パラメータの算出が必要となる．素性は本論文で述べた証拠自体となるため，素性の種類は頻度 7 で間引いて約 14 万弱

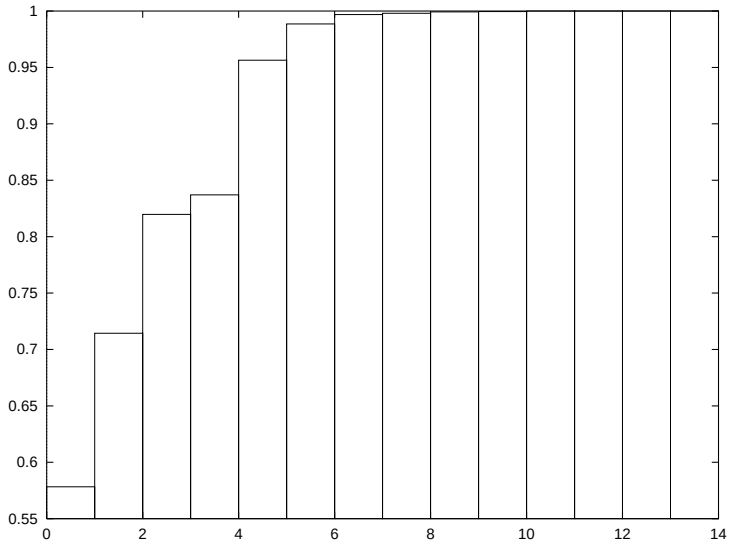

図 4　判別力と正解率
図 4　Identification strength and precision

である．最大エントロピー法で利用できる素性の数は現実的には，数万が限度であるために，最大エントロピー法の利用も現実的には無理がある．文字ベースの手法を利用する場合には，bi-gram や tri-gram などの情報を直接利用できる決定リストは現実的に有効な選択である．

本論文では単語分割を分類問題としてみなして解決した．分類問題とみなした場合，精度に関わる最も大きな要因は属性の選択である．アダブーストを利用するという枠組みでは，属性の設定はさらに考慮すべきである．ブースティングは弱学習アルゴリズムに対して利用できる．具体的には精度が 50% を越えるようなアルゴリズムであれば適用できる．つまり作成できた決定木などの分類システム自体の精度はそれほど高い必要はない．属性をうまく考慮して決定リストの精度を上げるよりも，作成される決定リストの精度は低いが，ブースティングにより精度が増してゆくような属性を設定するアプローチも有望である．いくつかの実験を行った結果，以下の点が確認できた．

- 属性を増やす，間引きの頻度を調整する，などの工夫を入れて決定リストの精度を上げた場合，ブースティングでは精度が上がらなかった．
- 属性を単純化して決定リストの精度を若干下げた場合，ブースティングによって精度は上がるが本実験で行った結果以上には精度は上がらなかった．

結論的には本論文で設定した属性の情報を利用する上では，本論文で示した値程度が限界に近いと感じられた．

分類誤りの原因を追求すると，訓練データに現れない表現あるいは頻度の低い表現の部分で

分類が誤っている[4]．これは未知語の問題そのものであり，未知語への対処が単語分割の中心の課題と言える．この解決策は3つ考えられる．1つ目は規則の一般化を精度良く行うことである．例えば文字クラス (小田, 森, 北 1999) などの導入などが考えられる．2つ目は別リソースの利用である．例えば辞書の利用である．単語分割に本手法の分類手法と辞書による最長一致法を利用することも考えられる．3つ目は訓練データの拡充である．事例ベースの手法 (山下 松本 1998; 伊東 1999) は訓練データつまり事例を大規模化することで精度が上がる．ただし大規模な正解付きの訓練データが用意できない現状では，正解のない訓練データをどう使うかが鍵となる (新納 2000)．1 つ目のアプローチ以外は，未知語の検出に対して理論的な保証がない．しかしだからといって，単語分割を文字ベースの手法によって解くことに意味がないわけではない．辞書に基づいた分割では数値表現や字種区切りが有効になるような未知語しか解析できず，解析できる未知語が限定されている．このような未知語の多くは，実験に示したように，本手法でもその多くを検出できる．さらに文字ベースの手法では，その他のタイプの未知語も検出できる場合が多々あるが，辞書に基づいた分割では確実に検出できない．この違いは大きい．

最後に本手法のアプローチは解析が決定的になるという長所もあることを付記しておく (Shinnou 2000)．通常の形態素解析システムも現実的にはほぼ文字数に比例した時間で解析が行えるので，決定的であるということはそれほど大きな長所ではない．ただし理論的に線形時間での解析を保証できることには意味がある．

6　おわりに

本論文では日本語単語分割を分類問題とみなし，決定リストを利用してその問題を解いた．本手法は未知語の問題を受けないという長所がある．実験では，文字ベースの n-gram モデルに基づく単語分割法との比較を行い，決定リストによる単語分割の方が優れていることを示した．また分類問題ととらえることで，ブースティングの手法を適用できることも示した．アダブーストを利用することによって，単独の決定リストよりもさらに精度を向上させることができた．未知語の検出能力も高かった．訓練データにない表現をどのようにカバーしてゆくかが今後の課題である．

謝辞

本研究は（ 財）栢森情報科学振興財団の研究助成金（ K11 研IV 第 71 号）によって行われました．深く感謝します.

4 先の実験により示した本手法が検出できなかった未知語の出現数（ 336）から考えて，全体の誤りの数（ 851）が多いようにも感じられる．しかしこれは，本実験では頻度7以下の証拠を間引いているために，本手法における未知語の実質的な総数は，先の実験で示した数よりも多いことによる．

参考文献

Freund, Y. and Schapire, R. E. (1997). "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of Computer and System Sciences*, **55** (1), 119–139.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publisher.

Ratnaparkhi, A. (1998). "Maximum Entropy Models for Natural Language Ambiguity Resolution." In *PhD thesis*. University of Pennsylvania.

Shinnou, H. (2000). "Deterministic Japanese Word Segmentation by Decision List Method." In *PRICAI-2000 (poster session)*, pp. 822–822.

Tsuji, K. and Kageura, K. (1997). "An HMM-based Method for Segmenting Japanese Terms and Keywords based on Domain-Specific Bilingual Corpora." In *The 4th Natural Language Processing Pacific Rim Symposium*, pp. 557–560.

Yarowsky, D. (1994). "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French." In *32th Annual Meeting of the Association for Computational Linguistics*, pp. 88–95.

伊東秀夫 (1999). "Suffix array を用いた日本語単語分割." 情報処理学会自然言語処理研究会, NL-131-7.

小田裕樹 北研二 (1998). "PPM^* モデルによる日本語単語分割." 情報処理学会自然言語処理研究会, NL-128-2.

小田裕樹, 森信介, 北研二 (1999). "文字クラスモデルによる日本語単語分割." 自然言語処理, **6** (7), 93–108.

北研二 (1999). 確率的言語モデル. 東京大学出版会.

新納浩幸 (2000). "日本語単語分割へのタグなしコーパスとタグ付きコーパスの利用." 情報処理学会自然言語処理研究会, NL-140-1.

山下達雄 松本裕治 (1998). "品詞タグ付きコーパスを直接利用した形態素解析." 言語処理学会第4回年次大会, pp. 524–527.

山本幹雄 増山正和 (1997). "品詞・区切り情報を含む拡張文字の連鎖確率を用いた日本語形態素解析." 言語処理学会第3回年次大会, pp. 421–424.

略歴

新納 浩幸: 昭和36年生．昭和60年東京工業大学理学部情報科学科卒業．昭和62年同大学大学院理工学研究科情報科学専攻修士課程修了．同年富士ゼロックス，翌年松下電器を経て，平成5年4月茨城大学工学部システム工学科助手．平成9年10月同学科講師，現在に至る．博士(工学)．

(2000 年 8 月 26 日 受付)
(2000 年 10 月 6 日 再受付)
(2001 年 1 月 12 日 採録)