統計的言語モデルとN-best探索を用いた日本語形態素解析法

永田昌明†

本論文では，統計的言語モデルとN-best探索アルゴリズムを用いた新しい日本語形態素解析法を提案する．本方法は，未知語の確率モデルを持つことにより任意の日本語文を高精度に解析し，確率が大きい順に任意個の形態素解析候補を求められる．EDRコーパスの部分集合（約19万文，約470万語）を用いて言語モデルの学習を行い，オープンテキスト100文に対してテストを行ったところ，単語分割の精度は第1候補で再現率94.6%適合率93.5%，上位五候補で再現率97.8%適合率88.3%であった．

A Japanese Morphological Analysis Method Using a Statistical Language Model and an N-best Search Algorithm

MASAAKI NAGATA†

We present a novel method for Japanese morphological analysis which uses a statistical language model and an N-best search algorithm. It has a probabilistic model for unknown words to parse unrestricted Japanese sentences accurately and it can get N-best morphological analysis hypotheses. When the statistical Japanese morphological analyzer was trained on the subset of the EDR corpus (about 190 thousand sentences, 4.7 million words) and tested on 100 sentences of open text, it achieved 94.6% recall and 93.5% precision for the top candidate, and 97.8% recall and 88.3% precision for the top five candidates.

1. はじめに

近年，大規模なテキストコーパスが利用可能になったことや，計算機の性能が大幅に向上したことから，自然言語の確率・統計的なモデルを作成する試みがさかんになってきた．確率的モデル化と機械学習に基づくアプローチは，広範な適用範囲（broad-coverage）を持ち，頑強（robust）で，高精度（accurate）な自然言語処理システムを構築できる可能性がある．

特に英語の品詞タグ付け（part of speech tagging）では，隠れマルコフモデル[3)~5),16)]や誤り主導の変換に基づく学習[2)]など，統計的言語モデルを利用する技術が急速に進歩した．これらの手法は95%以上の精度を持ち，人手で作成した文法規則を利用する従来手法よりも高精度かつ頑健なので，英語の品詞タグ付けの標準的な手法になっている．

これに対して日本語の形態素解析では，最長一致法[12)]，文節数最小法[25)]，接続コスト最小法[8)]など，人手で作成した文法規則と曖昧性解消のための発見的規則を組み合わせる手法が依然として主流である．特に接続コスト最小法は，日本語形態素解析の標準的な手法としてJUMAN[14)]，茶筅[13)]などのフリーソフトや市販の仮名漢字変換ソフトなどで使用されている．

しかし，接続コスト最小法で使用される品詞接続コストと単語コストの具体的な値は，試行錯誤により実験的に決定するしかなく，理論的根拠に乏しい．コスト設定には微妙なバランス感覚が要求され，ある領域（たとえば新聞）向けに調節したコストは，他の領域（たとえば特許文）では不適切なこともある．このように対象領域へのパラメータの適応や保守が難しいことが接続コスト最小法の問題点である．

本論文では，大規模コーパスにおける言語表現の出現頻度に基づいて日本語形態素解析用の統計的言語モデルを作成する新しい方法を提案する．この方法は，情報理論と確率論に基づく明確な理論的根拠を備え，対象領域のテキストからモデルのパラメータを学習する方法が存在し，かつ，95%程度の高い精度を持っている．簡潔にいえば，接続コスト最小法のコストを自動的に最適な値に設定することができる．

さらに本論文では，確率が大きい順番に1つずつ任意個の形態素解析候補を求めることができる日本語

† NTTサイバースペース研究所
NTT Cyber Space Laboratories

形態素解析用の N-best 探索法を提案する．接続コスト最小法はコスト最小（確率最大）の単語列しか求められない．しかし，たとえば仮名漢字変換の次候補のように，実用的には上位 N 個の最適解を求めたいことが多い．一般に上位 N 個の最適解を求めることを N-best 探索（N-best search）という[20]．本論文で提案する N-best 探索法は接続コスト最小法の自然な拡張になっており，上位 N 個の最適解（N をあらかじめ決める必要はない）を効率良く求めることができる．

以下では，まず統計的言語モデルについて説明し，次に N-best 探索法について説明する．続いて形態素解析精度の評価方法と実験結果について報告し，最後に考察と今後の課題を述べる．

2. 統計的言語モデル

2.1 日本語形態素解析の数学的定義

文字列 $C = c_1 \ldots c_m$ から構成される入力文が単語列 $W = w_1 \ldots w_n$ に分割されるとする．数学的には，日本語の形態素解析は与えられた文字列に対する単語列の条件付き確率 $P(W|C)$ を最大化する単語列 $\hat{W}$ を求める問題と定義できる．ここで文字列 C はすべての単語分割に共通なので $P(W)$ を最大化する単語列を求めればよい☆．

$$\hat{W} = \arg\max_W P(W|C) = \arg\max_W P(W) \quad (1)$$

本論文では，日本語の形態素解析において単語列の同時確率を計算するための統計的言語モデル $P(W)$ を単語分割モデル（word segmentation model）と呼ぶことにする．

単語分割モデルには，単語 ngram モデルや隠れマルコフモデルなど，音声認識や英語の品詞タグ付けに使われる統計的言語モデル[9),11)]と基本的に同じモデルを使用できる．ただし，英語と違って日本語は単語を分かち書きしないので，未知語（辞書未登録語）に関してはより洗練された確率モデルを必要とする．

本論文では未知語の確率モデルを単語モデル（word model）と呼ぶことにする．以下ではまず基本的な単語分割モデルについて説明し，次に日本語の単語モデルについて説明する，そして単語モデルを組み込んだ単語分割モデルについて説明する．

2.2 ngram モデル

まず最も簡単な言語モデルとして，マルコフモデル（ngram モデル）を説明する．$W = w_1 \ldots w_n$ の同時確率 $P(W)$ は次の条件付き確率の積に分解できる．

$$P(W) = \prod_{i=1}^{n} P(w_i|w_1 \ldots w_{i-1}) \quad (2)$$

さまざまな単語の組合せに対して条件付き確率 $P(w_i|w_1 \ldots w_{i-1})$ を推定することは現実的に不可能なので，これを $N-1$ 重マルコフ過程で近似したモデルを単語 ngram モデルという☆☆．

$$P(w_i|w_1 \ldots w_{i-1}) = P(w_i|w_{i-N+1} \ldots w_{i-1}) \quad (3)$$

右辺の単語 ngram 確率は，人手により単語分割された訓練テキストがあれば，そのテキスト中の単語列の相対頻度から推定できる．

$$P(w_i|w_{i-N+1} \ldots w_{i-1}) = \frac{C(w_{i-N+1} \ldots w_i)}{C(w_{i-N+1} \ldots w_{i-1})} \quad (4)$$

ここで，C は単語列の出現頻度を表す．N の値が大きいほど，訓練テキストから信頼性の高い単語 ngram 確率を推定するのが難しくので，英語の音声認識では $N = 2$ または $N = 3$ とすることが多く，それぞれ bigram，trigram と呼ばれる．$N = 1$ の場合，ngram 確率は単語の出現確率となるが，これは unigram と呼ばれる．

英語は分かち書きされているので訓練テキストを容易に入手できるが，日本語の場合，人手により単語分割されたテキストは，せいぜい数百万語程度しか利用可能ではない．そこで本論文では単語分割モデルとして単語 unigram および単語 bigram を用いる☆☆☆．

$$P(W) = \prod_{i=1}^{n} P(w_i) \quad (5)$$

$$P(W) = \prod_{i=1}^{n} P(w_i|w_{i-1}) \quad (6)$$

実際には，文頭および文末も特別な記号と見なした方が言語モデルの性能は向上する．たとえば単語 bigram は以下のようになる．

$$P(W) = P(w_1|\texttt{<bos>}) \prod_{i=2}^{n} P(w_i|w_{i-1}) \, P(\texttt{<eos>}|w_n) \quad (7)$$

☆ 本論文では，単語は表記・読み・品詞の 3 つ組から構成されると考える．2 つの単語はそれぞれの表記・読み・品詞がすべて一致するときに限り等しい．したがって，同形語（表記が同じ）や同音語（読みが同じ）は別々の単語と見なす．

☆☆ 一般に，ある事象の確率がその直前の N 個の事象だけに依存するとき，これを N 重マルコフ過程という．

☆☆☆ 日本語と同様に単語を分かち書きしない中国語の単語分割の研究では単語 unigram を用いるのが一般的である[21)]．

ここで <bos> および <eos> は文頭および文末を表す特殊記号である．

2.3 隠れマルコフモデル

次に隠れマルコフモデル（Hidden Markov Model, HMM）について説明する．隠れマルコフモデルは，観測不可能な（隠れた）マルコフ過程と，その状態に依存するシンボル生成器の組合せによって，シンボルの系列を表現するモデルである．

隠れマルコフモデルを言語モデルとして使用する場合，単語列 $W = w_1 \ldots w_n$ を観測可能なシンボル系列，品詞列 $T = t_1 \ldots t_n$ を観測不可能な状態系列と考え，$P(W)$ を品詞 bigram 確率 $P(t_i|t_{i-1})$ と品詞別単語出現確率 $P(w_i|t_i)$ の積で近似する☆．

$$P(W) = \prod_{i=1}^{n} P(t_i|t_{i-1})P(w_i|t_i) \tag{8}$$

英語の品詞付けでは，品詞 trigram 確率で文脈を表現する二次隠れマルコフモデルを用いることが多い．

$$P(W) = \prod_{i=1}^{n} P(t_i|t_{i-2}, t_{i-1})P(w_i|t_i) \tag{9}$$

式 (8) および式 (9) の言語モデルは，それぞれ品詞 bigram モデルおよび品詞 trigram モデルと呼ばれる．隠れマルコフモデルのパラメータは，人手により単語分割と品詞付与が行われた訓練テキストがあれば，対応する事象の相対頻度から求めることができる．

本論文では，日本語形態素解析の接続コスト最小法を確率化したものに相当する品詞 bigram モデル，および，英語の品詞付けで一般的に用いられている品詞 trigram モデルを単語分割モデルとして使用する．

2.4 単語モデル

任意のテキストを形態素解析するためには，未知語処理が不可欠である．英語は単語を分かち書きするので未知語を容易に同定できるが，べた書きされた日本語では入力文中の未知語を同定することが非常に難しい．

従来の未知語処理は，かなり「場あたり的」(ad hoc) である．「ひらがなから漢字に変化する部分は単語境界である可能性が高い」といった字種に関する発見的規則を用いる方法[26]や，解析に失敗した時点で付属語列などから文節境界を推定し，そこから付属語列を取り除いた部分列を未知語と見なす方法などがよく使われる．前者は，文字列の単語らしさを評価する方法，後者は，ある文脈における文字列の単語らしさを評価する方法と見なすことができるが，どちらの場合も，尤度の根拠が不明確であり，単語候補の詳細な順位付けも難しい．

そこで本論文で提案する言語モデルでは，任意の文字列に対して単語出現確率（単語としてのもっともらしさ）を割り当てる単語モデルを定義し，ある文脈における未知語の出現確率を与えるように単語分割モデルを拡張する．

数学的には，ある単語 w_i が未知語であるとき，その表記が長さ k の文字列 $c_1 \ldots c_k$ である確率 $P(c_1 \ldots c_k|\texttt{<UNK>})$ の計算モデルを単語モデルと定義する．ここで <UNK> は未知語を表す記号である．

$$P(w_i|\texttt{<UNK>}) = P(c_1 \ldots c_k|\texttt{<UNK>}) \tag{10}$$

最も簡単な単語モデルの 1 つは，未知語の出現確率を各文字の出現確率の積で近似する方法である．

$$P(c_1 \ldots c_k|\texttt{<UNK>}) = \prod_{i=1}^{k} P(c_i) \tag{11}$$

しかし，この単語モデルには長い未知語の出現確率が非常に小さくなるという問題がある．また，文字種の変化点が単語境界になることが多いという日本語の特徴を十分に表現できない．そこで本論文では単語長と文字 bigram を利用する単語モデルを提案する．

単語モデルは，一般性を失うことなく，未知語の文字長の分布を表す単語長確率と，ある文字長の未知語の表記の出現確率を表す単語表記確率の積に分割できる．

$$P(c_1 \ldots c_k|\texttt{<UNK>}) = P(k|\texttt{<UNK>})\, P(c_1 \ldots c_k|k, \texttt{<UNK>}) \tag{12}$$

単語長確率 $P(k|\texttt{<UNK>})$ は平均単語長 λ をパラメータとするポワソン分布に従うと仮定する．これは，長さ 0 の単語区切り記号を考え，平均間隔が平均単語長に等しくなるように区切り記号をランダムに配置するモデルで単語分割を近似することを意味する．

$$P(k|\texttt{<UNK>}) = \frac{(\lambda - 1)^{k-1}}{(k-1)!} e^{-(\lambda-1)} \tag{13}$$

単語表記確率 $P(c_1 \ldots c_k|k, \texttt{<UNK>})$ は，単語内文字 bigram モデルから求めた文字列 $c_1 \ldots c_k$ の出現確率 $P_b(c_1 \ldots c_k)$ と，単語内文字 bigram モデルにおいて長さ k の文字列が出現する確率 $P_b(k)$ の比で近似する

$$P(c_1 \ldots c_k|k, \texttt{<UNK>}) = \frac{P_b(c_1 \ldots c_k)}{P_b(k)} \tag{14}$$

☆ 隠れマルコフモデルにおける品詞 bigram 確率および品詞別単語出現確率の対数の絶対値を，接続コスト最小法における品詞接続コストおよび単語コストに対応付ければ，前者の最尤解は後者のコスト最小解に対応する．したがって，隠れマルコフモデルは情報理論的な根拠を持った接続コストのモデルと見なせる．

$$P_b(c_1 \dots c_k) = P(c_1|\texttt{<bow>}) \prod_{i=2}^{k} P(c_i|c_{i-1}) P(\texttt{<eow>}|c_k) \quad (15)$$

$$P_b(k) = (1 - P(\texttt{<eow>}))^{k-1} P(\texttt{<eow>}) \quad (16)$$

ここで <bow>および <eow> は語頭および語末を表す特殊記号である．単語内文字bigram確率は，単語の先頭（接頭辞）・中間・末尾（接尾辞）に現れる文字bigramでは大きく，単語境界をはさむ文字bigramでは小さくなる．したがって，$P_b(c_1 \dots c_k)$ は単語を構成する文字列では大きく，そうでない文字列では小さくなるので，かなり良い単語モデルになる．

しかし，$P_b(c_1 \dots c_k)$ がすべての長さの文字列の中で $c_1 \dots c_k$ が出現する確率なのに対して $P(c_1 \dots c_k|k, \texttt{<UNK>})$ は長さ k の文字列における出現確率なので，(未知語か否かは無視して) 前者で後者を近似すると長い文字列の確率が不適切に小さな値になる．そこで式(14)のように長さ k の文字列の出現確率 $P_b(k)$ で割って補正する必要がある．$P_b(k)$ は式(16)に示すように語末記号 <eow> 以外の文字が $k-1$ 個続いた後に語末記号が出現する事象の確率から求められる．

2.5 未知語を考慮した単語分割モデル

次に単語モデルを使って，未知語を考慮した単語分割モデルを定義する．まず隠れマルコフモデルでは，未知語を特別な品詞と考え，未知語クラス <UNK> の品詞bigram確率で未知語が出現する確率を表す．品詞別単語出現確率，すなわち表記が w_i である未知語の出現確率は，定義により単語モデルに等しい．

$$P(t_i|t_{i-1}) = P(\texttt{<UNK>}|t_{i-1}) \quad (17)$$

$$P(w_i|t_i) = P(w_i|\texttt{<UNK>}) \quad (18)$$

品詞trigramモデルの場合も同様である．

単語unigramモデルでは，未知語 w_i の単語出現確率は未知語クラスの出現確率と単語モデルの積に等しい．

$$P(w_i) = P(\texttt{<UNK>}) P(w_i|\texttt{<UNK>}) \quad (19)$$

単語bigramモデルでは，未知語 w_i の単語bigram確率は未知語クラスの単語bigram確率と単語モデルの積に等しい．

$$P(w_i|w_{i-1}) = P(\texttt{<UNK>}|w_{i-1}) P(w_i|\texttt{<UNK>}) \quad (20)$$

未知語クラスの出現確率 $P(\texttt{<UNK>}|t_{i-1})$，$P(\texttt{<UNK>})$，$P(\texttt{<UNK>}|w_{i-1})$ は，すべての辞書未登録語を未知語記号 <UNK> に置き換えた学習コーパスから推定する．この方法は，低頻度単語（系列）に割り当てられた確率を未知語（未知系列）に再配分するという点では，バックオフ法[10]におけるグッド・チューリング推定[6]と基本的に同じであるが，単語モデルに基づいて未知語全体に割り当てられた確率を再配分する部分に新規性がある．

3. N-best探索アルゴリズム

本章では，確率が大きい順番に1つずつ任意個の形態素解析候補を求める新しい探索法（前向きDP後向き A^* アルゴリズム）を提案する☆．このアルゴリズムは，文頭から文末へ1文字ずつ進む動的計画法（DP, Dynamic Programming）を用いて文頭から任意の単語までの部分解析の確率を求める前向き探索と，文末から文頭へ進む A^* アルゴリズムを用いて確率が大きい順に形態素解析候補を求める後向き探索から構成される．

以下では，まず単語分割モデルに単語bigramを用いる場合を例として前向き探索と後向き探索のアルゴリズムを説明し，次に高次マルコフモデル（trigram以上）に本アルゴリズムを適用する方法を説明する．

3.1 前向き探索

前向き探索では，式(1)の解，すなわち，同時確率 $P(W)$ を最大化する単語列 $\hat{W}$ を動的計画法により求める．文頭から i 番目の単語までの単語列の同時確率 $P(w_1 \dots w_i)$ の最大値を $\phi(w_i)$ と定義すると，式(6)より，以下の関係が成立する．

$$\phi(w_i) = \max_{w_{i-1}} \phi(w_{i-1}) P(w_i|w_{i-1}) \quad (21)$$

すなわち，文頭から i 番目の単語までの同時確率の最大値 $\phi(w_i)$ は，文頭から $i-1$ 番目の単語までの同時確率の最大値 $\phi(w_{i-1})$ と単語bigram確率 $P(w_i|w_{i-1})$ の積の最大値である．この関係を用いて文頭から順番に $\phi(w_i)$ を求めれば，文頭から文末までの同時確率の最大値 $\phi(w_n)$ を求めることができる．

動的計画法を用いて式(21)の計算を実現するアルゴリズムを**図1**に示す．長さ m の日本語文字列を $C = c_1 \dots c_m$ とし，部分文字列 $c_{p+1} \dots c_q$ を c_p^q で表す．辞書 D の中で表記が文字列 c_q^r と等しい単語の集合を $D(c_q^r) = \{w_i | w_i = c_q^r, w_i \in D\}$ で表すことにすると，入力文の文字位置 q から文末までの部分文

☆ 本論文のN-best探索アルゴリズムは，文献18)で提案したアルゴリズムをより洗練された形に改訂したものである．文献18)では言語モデルとして品詞ngramを仮定し，アルゴリズムの記述に複雑な構造体やリスト演算を必要としたが，本論文では言語モデルとして単語bigramを使用し，アルゴリズムの記述には簡単な集合演算と必要最小限のテーブルしか使用しない．

```
1   T_0 ← {w_0}
2   φ(w_0) ← 1
3   for q = 0 to m do
4    foreach w_{i-1} ∈ T_q do
5     foreach w_i ∈ ∪_{q<r≤m} D(c_q^r) do
6      if w_i ∉ T_r then
7       T_r ← T_r ∪ {w_i}
8       φ(w_i) ← 0
9      endif
10     if (φ(w_{i-1}) P(w_i|w_{i-1}) > φ(w_i)) then
11      φ(w_i) ← φ(w_{i-1}) P(w_i|w_{i-1})
12     endif
13    end
14   end
15  end
```

図1 動的計画法を用いた前向き探索アルゴリズム
Fig. 1 Forward search algorithm.

字列の接頭辞と一致する単語の集合は $\cup_{q<r\leq m} D(c_q^r)$ と表せる．また，文字位置 q で終わる単語を記憶するテーブルを $T_q = \{w_{i-1} | w_{i-1} = c_p^q, 0 \leq p < q\}$ で表す．

図1のアルゴリズムは，現在の文字位置を q で表し，文頭から文末方向へ1文字ずつ進む（3行目から15行目までのfor文）．まず文頭を表す特殊な記号 w_0 を文字位置0で終わる単語のテーブル T_0 に格納し，最適単語列確率 $\phi(w_0)$ を1に初期化する（1，2行目）．各文字位置 q では，その文字位置で終わる確率最大の単語列（4行目のfor文）とその文字位置から始まる単語（5行目のfor文）を組み合わせて新しい単語列を作成し，もし新しい単語列の確率が以前の単語列の確率よりも大きければ，最適単語列の確率を更新する（6行目から12行目まで）☆．

前向き探索の計算量を支配するのは一番外側のfor文（3行目）であり，4行目から14行目までの計算は各文字位置で1度だけ行われるので，前向き探索の計算量は文の文字数に比例する．

図2に，“会議に申し込みたい”という入力文に対する文字位置7における動的計画法の様子を示す．品詞の違いも考慮すると，この文字位置で終わる単語が4個，この文字位置から始まる部分文字列と一致する単語が4個ある．これらのすべての組合せを調べ，単語列の終わりの文字位置の最適単語列確率を更新する．

☆ 具体的には，文字位置 q から始まる単語 w_i の終了位置を r とし，単語 w_i がテーブル T_r に未登録ならば，これを登録して，最適単語列確率 $\phi(w_i)$ の初期値を0とする（6行目から9行目）．もし w_{i-1} から w_i へ至る単語列の確率 $\phi(w_{i-1})P(w_i|w_{i-1})$ が，w_i へ至る単語列の確率のそれまでの最大値 $\phi(w_i)$ より大きければ，$\phi(w_i)$ を更新する（10行目から12行目）．

3.2 後向き探索

後向き探索では，人工知能の分野において状態空間グラフの最小コスト経路を求めるアルゴリズムである A^* アルゴリズムを用いて式(1)の確率が大きい順に1つずつ形態素解析候補を求める．

まず A^* アルゴリズムについて簡単に説明する．グラフの任意のノードを n としたとき，初期状態から n までの最適な経路のコストを $g(n)$ とし，n から最終状態までの最適な経路のコストを $h(n)$ とすれば，n を通る最適な経路のコスト $f(n)$ は次式で与えられる．

$$f(n) = g(n) + h(n) \tag{22}$$

もし $f(n)$ が正確に分かれば，初期状態から $f(n)$ が最小となるノードをたどることにより最終状態への最適な経路を求められるが，一般に $f(n)$ は正確には分からない．

$g(n)$ をそれまで分かっている n までの最適経路のコストとし，$h(n)$ の推定値を $\hat{h}(n)$ とする．もし推定コスト $\hat{h}(n)$ が真のコスト $h(n)$ より小さければ，すなわち $\hat{h}(n) \leq h(n)$ ならば，$\hat{f}(n) = g(n) + \hat{h}(n)$ が最小となるノードをたどることにより最適解が得られることを証明できる．この性質を利用したグラフ探索戦略を A^* アルゴリズムという．また，ある探索アルゴリズムが必ず最適解を発見できるとき，その探索アルゴリズムは認容可能（admissible）であるという．

特に推定コスト $\hat{h}(n)$ と真のコスト $h(n)$ が一致する場合，A^* アルゴリズムは（最適経路以外のノードを通ることなく）ただちに最適経路を求められる．一般に推定コストが真のコストに近いほど，探索量は少ない．

次に A^* アルゴリズムの後向き探索への適用法を説明する．ある形態素解析候補 $W = w_1 \ldots w_n$ において，i 番目の単語が w_i という条件の下で，$i+1$ 番目の単語から文末までの単語列の同時確率 $P(w_{i+1} \ldots w_n | w_i)$ を $\psi(w_i)$ と定義すると，前向き探索の場合と同様に，式(6)より，以下の関係が成立する．

$$\psi(w_i) = P(w_{i+1}|w_i)\psi(w_{i+1}) \tag{23}$$

$\phi(w_i)$ を前向き部分解析の確率，$\psi(w_i)$ を後向き部分解析の確率と呼ぶことにする．ある形態素解析候補 W の確率は，任意の w_i について，前向き部分解析の確率と後向き部分解析の確率の積で表せる．

$$P(W) = \phi(w_i)\psi(w_i) \tag{24}$$

後向き探索では，関数 $g(n)$ として後向き部分解析の確率 $\psi(w_i)$ の対数の絶対値，関数 $h(n)$ として前向き部分解析の確率 $\phi(w_i)$ の対数の絶対値を用いて A^* アルゴリズムを適用する．前向き探索により $h(n)$ の真の値が分かっているので，後向き探索は必ずかつた

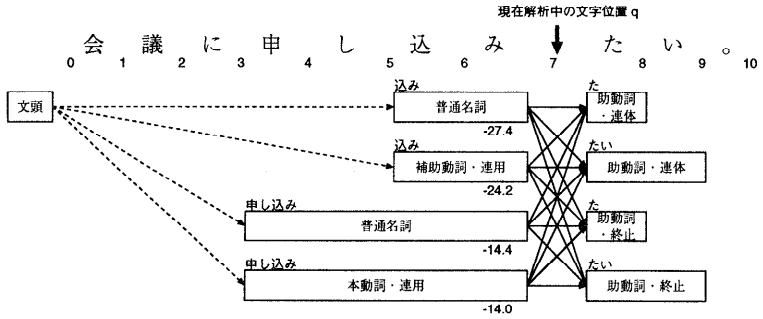

図 2 動的計画法を用いた前向き探索
Fig. 2 Forward search using dynamic programming.

だちに最適解を得られる．最適解が得られたら，そのノードを取り除き，さらに探索を続けることにより次の最適解が得られる．したがって，後向き探索は認容可能で，正確かつ効率的に上位 N 個の形態素解析候補を求めることができる．

A^* アルゴリズムを用いた後向き探索のアルゴリズムを図 **3** に示す．一般に A^* アルゴリズムでは open と closed という 2 つのリストを用いる．リスト open には，すでに生成され，残りの経路のコスト $h(n)$ を求めたが，まだ展開されていないノードの集合を格納し，リスト closed にはすでに展開されたノードの集合を格納する．

後向き探索では，まず文末を表す特殊記号 w_{n+1} を open に代入し，closed には空リストを代入する（1, 2 行目）．後向き探索のコスト g の初期値は 0 であり，全経路のコスト f の初期値は前向き探索のコスト $h(w_{n+1})$ とする（3, 4 行目）．そして open の要素が空になるか（5 行目），または探索が文頭に達する（7 行目）まで open の要素を 1 つずつ展開する（5 行目から 35 行目まで）．

後向き探索の各ステップでは，open の中で全経路のコスト f が最も小さい単語 w_i を選び（6 行目），これを open から closed へ移動し（8, 9 行目），w_i と後向きに接続可能なすべての単語について後向き探索のコストを計算し，必要に応じて open と closed を修正する（10 行目から 33 行目）．ここで $T_{start(w_i)}$ は w_i の開始位置を終了位置とする単語の集合を表す．また w_i の開始位置で接続可能な単語を w_{i-1} で表す．

文末から w_i を経由して w_{i-1} に至る後向き探索のコスト $g'(w_i, w_{i-1})$ は，文末から w_i までのコスト $g(w_i)$ と，w_i から w_{i-1} への遷移のコスト $|\log P(w_i|w_{i-1})|$ の和である（11 行目）．また文末から w_i と w_{i-1} を経由して文頭へ至る全経路のコスト $f'(w_i, w_{i-1})$ は，$g'(w_i, w_{i-1})$ と，前向き探索で求めた文頭から w_{i-1} までのコスト $h(w_{i-1})$ の和である（12 行目）．

もし，w_{i-1} が open に含まれており，w_i を経由して w_{i-1} に至る経路のコストが以前の経路のコストよりも小さければ，w_i を経由する経路およびコストを記録する（13 行目から 18 行目）．ここで $q(w_{i-1})$ は

```
1   open ← {w_{n+1}}
2   closed ← φ
3   g(w_{n+1}) ← 0
4   f(w_{n+1}) ← h(w_{n+1})
5   while open ≠ φ do
6     w_i ← arg min_{w∈open} f(w)
7     if w_i = w_0 then return SUCCESS
8     open ← open − {w_i}
9     closed ← closed ∪ {w_i}
10    for w_{i−1} ∈ T_{start(w_i)} do
11      g'(w_i, w_{i−1}) ← |logP(w_i|w_{i−1})| + g(w_i)
12      f'(w_i, w_{i−1}) ← g'(w_i, w_{i−1}) + h(w_{i−1})
13      if w_{i−1} ∈ open then
14        if f'(w_i, w_{i−1}) < f(w_{i−1}) then
15          g(w_{i−1}) ← g'(w_i, w_{i−1})
16          f(w_{i−1}) ← f'(w_i, w_{i−1})
17          q(w_{i−1}) ← w_i
18        endif
19      else if w_{i−1} ∈ closed then
20        if f'(w_i, w_{i−1}) < f(w_{i−1}) then
21          g(w_{i−1}) ← g'(w_i, w_{i−1})
22          f(w_{i−1}) ← f'(w_i, w_{i−1})
23          q(w_{i−1}) ← w_i
24          closed ← closed − {w_{i−1}}
25          open ← open ∪ {w_{i−1}}
26        endif
27      else
28        g(w_{i−1}) ← g'(w_i, w_{i−1})
29        f(w_{i−1}) ← f'(w_i, w_{i−1})
30        q(w_{i−1}) ← w_i
31        open ← open ∪ {w_{i−1}}
32      endif
33    end
34  end
35  return FAILURE
```

図 3 A^* アルゴリズムを用いた後向き探索アルゴリズム
Fig. 3 Backward search algorithm.

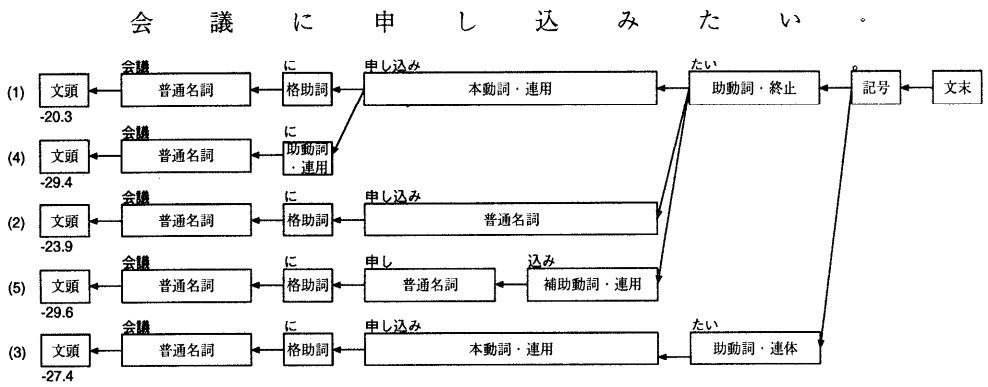

図4 A^* アルゴリズムを用いた後向き探索
Fig. 4 Backward search using A^* algorithm.

後向き探索において w_{i-1} の直前の状態を記録するテーブルを表す．もし，w_{i-1} が closed に含まれており，w_i を経由して w_{i-1} に至る経路のコストが以前の経路のコストよりも小さければ，w_i を経由する経路およびコストを記録するとともに，w_i を closed から open へ移動する（19 行目から 26 行目）．w_{i-1} が open にも closed にも含まれていなければ，w_i を経由する経路およびコストを記録するとともに，w_{i-1} を open に加える（27 行目から 32 行目）．

後向き探索の計算量を支配するのは一番外側の while 文（5 行目から 34 行目）の実行回数であるが，これは必要最小回数しか実行されない．すなわち，第 1 候補を求める際には単語数と同じ回数だけ実行され，第 2 候補以下では上位候補と異なる単語が選択されたところから文頭までの単語数と同じ回数だけ実行される．なぜなら全経路コスト f が前向き探索によりあらかじめ分かっているので，6 行目で選んだ全経路コスト最小の単語 w_i と同じ全経路コストを持つ単語が，10 行目で求めた w_i に接続可能な単語集合 $T_{start(w_i)}$ の中に必ず存在し，それが必ず次に選ばれるからである．

従来の接続コスト最小法でも，求めるべき形態素解析候補の数 N をあらかじめ決め，文頭からある単語までの経路をコストが小さい順につねに N 個ずつ記録すれば，上位 N 個の候補を求めることができる[8]．しかし，この方法は，文頭からある単語までのコストの最小値（確率の最大値）だけを記録する本論文の前向き探索に比べると，計算量も記憶量も N 倍になる．また本論文の N-best 探索法はあらかじめ候補数 N を決める必要がない．

図 4 に，“会議に申し込みたい.” という文の後向き探索の様子を示す．この図において，左端の数字が候補順位であり，左端の箱の下の数字は全経路の対数確率である．この図より，「申し込み（本動詞連用形と普通名詞）」「たい（助動詞終止形と助動詞連用形）」「に（格助詞と助動詞連用形）」に関する品詞の多義，および，「申し込み」と「申し | 込み」という単語分割の多義が最ももっともらしい順番に提示される様子や，必要最小限のノードしか展開されない様子が分かる．

3.3 高次マルコフモデルへの適用

ここでは前向き DP 後向き A^* アルゴリズムを高次（隠れ）マルコフモデルへ適用する方法を説明する．一般に N 次マルコフモデルは直前の N 個の状態を 1 つの複合状態（combined state）と見なすことにより一次マルコフモデル（bigram）に変換できる．ただし，文字 ngram モデルにおける拡張ビタビアルゴリズム（extended Viterbi algorithm）[7] と比較すると，日本語の形態素解析は出力シンボル（単語）の重なりと内部状態（品詞）の違いを考慮する必要があるので，変換が少し複雑になる．

以下では品詞 trigram モデルを例として説明する．品詞 trigram モデルの場合，$u_1 = t_1$ かつ $u_i = t_{i-1}t_i$ として，複合状態系列 $U = u_1 u_2 \ldots u_n$ を考えると次の関係が成り立つ．

$$P(u_i|u_{i-1}) = P(t_i|t_{i-2}, t_{i-1}) \tag{25}$$

式 (25) を式 (9) へ代入すると次式が得られる．

$$P(W) = \prod_{i=1}^{n} P(u_i|u_{i-1})P(w_i|t_i) \tag{26}$$

式 (26) は，品詞 bigram モデル（一次隠れマルコフモデル）と同じ形である．ここで，文頭から i 番目の単語までの単語列 $W_i = w_1 \ldots w_i$ の同時確率 $P(W_i)$ の各 u_i ごとの最大値を $\phi(u_i)$ とすると，次式が得られる．

$$\phi(u_i) = \max_{u_{i-1}} \phi(u_{i-1})P(u_i|u_{i-1})P(w_i|t_i) \tag{27}$$

したがって，図 1 の前向き探索アルゴリズムにおい

て，単語を複合状態に置き換え，11 行目の確率の計算を式 (27) に変更すれば，文頭から文末までの同時確率の最大値を求めることができる．

後向き探索も，基本的には図 3 の後向きアルゴリズムで単語を複合状態に置き換えればよい．ただし，複合状態の遷移には特別な制約がある．単語の場合は文末側の単語の開始位置と文頭側の単語の終了位置が同じであれば遷移できるが（10 行目），複合状態の場合はさらに文末側の複合状態から最後の要素を取り除いた系列が，文頭側の複合状態から最初の要素を取り除いた系列と一致しなければ遷移できない．

4. 実　　験

4.1 言語データ

本論文では形態素解析プログラムの訓練と試験のために「EDR 日本語コーパス Version 1.0」[19] を用いた．EDR コーパスは，新聞・雑誌・辞書・百科事典・教科書などから収集され，形態論・統語論・意味論レベルのさまざまな注釈が人手で付与された約 500 万語（約 20 万文）のコーパスである．この実験では単語区切り・読み・品詞の情報を用いた．

まずコーパス全体の約 90%に相当する文を無作為に抽出して訓練集合とし，残りの 10%の中からテスト集合（100 文）を無作為に抽出した．**表 1** に訓練集合とテスト集合の文・単語・文字の数を示す．

訓練テキスト中の異なり単語数は 133281 個であり，頻度 2 以上の 65152 単語を辞書に登録した．そして頻度 1 の単語を未知語記号 <UNK> で置き換えた訓練テキストから単語 unigram，単語 bigram，品詞 bigram，品詞 trigram の 4 種類の単語分割モデルを作成した．

単語 unigram モデルは頻度 2 以上の単語に文末記号と未知語記号を加えた 65154 個の単語 unigram を使用した．単語 bigram モデルはすべての単語 bigram（758172 個）のうち頻度 2 以上の 294668 個を使用した．品詞 bigram モデルはすべての品詞 bigram（259 個）を使用した．品詞 trigram モデルはすべての品詞 trigram（2389 個）を使用した☆．

単語モデルについては，訓練テキスト中の異なり文字数は 3534 個であり，頻度 2 以上の 3167 個を既知文字とした．頻度 1 の文字を未知記号タグに置き換えた訓練テキストから単語内文字 bigram を求め，頻度 2 以上の 91198 個の文字 bigram を単語表記確率の計算に使用した．単語長確率の計算には，高頻度語の影響を避けるために頻度 1 の単語の平均単語長 4.76 を使用した．なおすべての ngram 確率は削除補間法により平滑化した[9]．

☆ EDR コーパスでは，名詞・動詞・形容詞・形容動詞・副詞・連体詞・接続詞・接頭語・接尾語・語尾・助詞・助動詞・感動詞・記号・数字の 15 個の品詞が用いられている．

表 1 訓練データと試験データの量
Table 1 The amount of the training and test data.

	訓練集合	テスト集合
文	192802	100
単語	4746461	2463
文字	7521293	3912

4.2 形態素解析の評価尺度

英語と違って日本語は単語を分かち書きしないので，止書法の中に単語という単位が存在しない．また日本語は膠着語なので，単語境界を一貫性を保ちながら決定するのは日本人にとっても難しい．このため，これまで日本語の形態素解析には標準的な評価尺度が存在しなかった．本論文では，英語の構文解析の評価尺度である括弧付け精度（bracketing accuracy）[1] を参考にして日本語の形態素解析の評価尺度を新たに定義し，さらに N-Best 探索の精度を表現できるように拡張する[18]．

形態素解析の精度は，人手で作成した正解の単語列と形態素解析システムが出力した単語列を比較し，単語が正しく同定されている割合を再現率（recall）と適合率（precision）で表現する．正解データに含まれる単語数を Std，システム出力に含まれる単語数を Sys，一致した単語数を M とすると，再現率は正解データの単語列の中でシステムが正しく同定した単語の割合 M/Std を表し，適合率はシステムが出力した単語列の中で正しく同定された単語の割合 M/Sys を表す．

一致した単語数を求める際には，形態素解析候補を単語の区切り・読み・品詞からなる 3 つ組の集合と考え，単語分割・品詞付け・形態素解析などの評価したい項目に応じて以下の 3 つの等価性基準を適用する．

単語分割　単語の区切りが等しい．

品詞付け　単語の区切りと品詞が等しい．

形態素解析　単語の区切りと読みと品詞が等しい．

N-best 候補の精度は，上位 N 個の単語列候補に含まれる単語の集合和を求め，これを正解データに含まれる単語の集合と比較し，再現率と適合率を求める．また，形態素解析の精度を 1 つの指標で表現したい場合には，情報検索で用いられる F-尺度を使用する．

$$F = \frac{(\beta^2 + 1.0) \times P \times R}{\beta^2 \times P + R} \tag{28}$$

ここで P は再現率，R は適合率，β は適合率に対する再現率の相対的重要度を表す．本論文では $\beta = 1.0$ とし，再現率と適合率の重みを等しくした．

EDR-00050001F8C0
5倍の割り増し合格でこれも全国公立大の中で最高となった.

正解データ		システム出力
5/5/数字		5/5/数字
倍/バイ/接尾語		倍/バイ/接尾語
の/ノ/助詞		の/ノ/助詞
割り増し合格/ワリマシゴウ		割り増し合格/ワリマシゴウ
で/デ/助動詞	|	で/デ/助詞
これ/コレ/名詞		これ/コレ/名詞
も/モ/助詞		も/モ/助詞
全/ゼン/名詞	|	全/ゼン/接頭語
国公立/コッコウリツ/名詞	|	国公立大/コッコウリツダイ
大/ダイ/名詞	<	
の/ノ/助詞		の/ノ/助詞
中/ナカ/名詞		中/ナカ/名詞
で/デ/助詞		で/デ/助詞
最高/サイコウ/名詞		最高/サイコウ/名詞
と/ト/助詞		と/ト/助詞
な/ナ/動詞		な/ナ/動詞
っ/ッ/語尾		っ/ッ/語尾
た/タ/助動詞		た/タ/助動詞
./. /記号		./. /記号

図5 形態素解析の例
Fig. 5 An example of morphological analysis.

例として「5倍の割り増し合格でこれも全国公立大の中で最高となった.」という文の形態素解析における正解データとシステム出力を**図5**に示す.正解データとシステム出力を比較すると,「国公立大」という部分文字列の単語分割,および,「で」と「全」の品詞が異なっている.正解データに含まれる単語は19個,システム出力に含まれる単語は18個であり,両者の中で区切りが等しいものは17個,区切りと品詞が等しいものは15個,区切りと読みと品詞が等しいものは15個である.したがって,単語分割の再現率と適合率は17/19と17/18であり,品詞付けおよび形態素解析の再現率と適合率は15/19と15/18である.

4.3 単語分割モデルと単語分割精度

単語unigram,単語bigram,品詞bigram,品詞trigramの4つの単語分割モデルについて,テスト集合(100文)に対する単語分割精度および品詞付け精度を**表2**および**表3**に示す.これらの表には各言語モデルのテストセットパープレキシティ(test set perplexity)[9),11)]も示した.このパープレキシティの値は,未知語を含むngramに単語分割モデルと単語モデルが割り当てる確率を考慮している.未知語を含むngramを対象外とした場合,単語unigram,単語bigram,品詞bigram,品詞trigramのパープレキシティは,それぞれ494,106,291,253である.

単語分割および品詞付けの精度は,品詞trigramや単語unigramよりも単語bigramの方が明らかに高い.また一般にパープレキシティが小さいほど単語分割および品詞付けの精度が高い☆.これは音声認識におけるパープレキシティと認識精度の関係と同じである.したがって,音声認識だけでなく形態素解析においてもパープレキシティは言語モデルの評価尺度として有効であり,パープレキシティの小さな言語モデルを設計することが形態素解析精度の向上に大きく貢献することが分かる.

表2 単語分割モデルと単語分割精度
Table 2 Word segmentation model and word segmentation accuracy.

	再現率	適合率	F-尺度	perplexity
単語 unigram	89.0	91.2	90.1	865
単語 bigram	94.6	93.5	94.1	191
品詞 bigram	91.5	92.9	92.2	500
品詞 trigram	91.5	92.5	92.0	452

表3 単語分割モデルと品詞付け精度
Table 3 Word segmentation model and tagging accuracy.

	再現率	適合率	F-尺度	perplexity
単語 unigram	80.6	82.6	81.6	865
単語 bigram	91.9	90.9	91.4	191
品詞 bigram	87.5	88.8	88.2	500
品詞 trigram	88.1	89.0	88.5	452

表2において,単語unigramモデルによる単語分割精度が約90%であることは注目に値する.単語分割では,単語の出現確率の情報が最も重要であり,品詞および単語の連接に関する情報は二次的なものである.この事実は未知語対策の重要性を示唆する.

4.4 単語モデルと単語分割精度

未知語処理における単語モデルの有効性を検証するために,以下のような比較実験を行った.まず単語モデルとして,最も単純なモデルと本論文で提案するモデルを用意する.前者は式(11)に示した文字unigram確率から未知語の出現確率を推定するモデルであり,後者は式(12)に示した平均単語長と文字bigram確率から未知語の出現確率を推定するモデルである.

次に単語分割モデルとして,辞書登録語彙数の異なる2つの単語bigramモデルを用意する.1つは訓練テキスト中で頻度2以上の単語(65152個)を使用する単語bigramモデル(前節と同じ)であり,もう1つは頻度41以上の単語(6923個)を辞書に登録し,頻度40以下の単語を未知語記号<UNK>で置き換えた訓練テキストから求めた単語bigramのうち頻度21以

☆ 品詞bigramによる単語分割のF-尺度は品詞trigramより高いが,これはたまたま品詞bigramの出力単語数が少ないために適合率が高くなっただけで,再現率は同じである.品詞付けのF-尺度は品詞trigramの方が高い.

表 4 単語モデルと単語分割精度
Table 4 Word model and word segmentation accuracy.

辞書	単語モデル	再現率	適合率	F-尺度
65 K	文字 unigram	94.8	93.2	94.0
65 K	単語長+文字 bigram	94.6	93.5	94.1
7 K	文字 unigram	93.3	88.9	91.0
7 K	単語長+文字 bigram	94.2	92.1	93.1

上の 24565 個を使用する単語 bigram モデルである．

テスト文（100 文）に対する 2 つの単語分割モデルの未知語率は，語彙数の大きい単語 bigram モデル（65 K 語）が 2.6%（65/2463），語彙数の小さい単語 bigram モデル（7 K 語）が 10.8%（266/2463）である．したがって，語彙数の小さい単語分割モデルでは，単語モデルに基づく未知語処理の成否が単語分割精度に大きく影響する．

語彙数の異なる単語分割モデルに対する 2 つの単語モデルの単語分割精度を**表 4** に示す．語彙数の大きい単語分割モデルを使用する場合，すなわち，テスト文の未知語率が小さい場合には，平均単語長と文字 bigram 確率を用いる単語モデルの方が文字 unigram 確率のみを用いる単語モデルよりもわずかに精度が高いが，その差は有意ではない．しかし，語彙数が小さい単語分割モデルを使用する場合，すなわち，未知語率が大きい場合には，平均単語長と文字 bigram 確率を用いる単語モデルの方が明らかに優れている．

一般に，未知語率が大きくなるのは，学習データが少ない場合，言語モデルを学習したテキストと異なる分野のテキストを扱う場合，あるいは，文字認識や音声認識による入力誤りを含むテキストを扱う場合などである．したがって，本論文で提案する単語モデルは形態素解析の頑健性の向上に大きく貢献することが分かる．

4.5 N-best 形態素解析の精度

単語の区切り・読み・品詞に関する N-best 候補の再現率と適合率を**図 6** に示す．単語分割（区切り）の精度は，第 1 候補で再現率 94.6%適合率 93.5%，上位 5 候補で再現率 97.8%適合率 88.3%である．品詞付け（区切りと品詞）の精度は，第 1 候補で再現率 91.9%適合率 90.9%，上位 5 候補で再現率 95.9%適合率 80.2%である．形態素解析（区切りと読みと品詞）の精度は，第 1 候補で再現率 91.7%適合率 90.6%，上位 5 候補で再現率 95.7%適合率 79.3%である☆．

☆ 単語分割の精度に比べて品詞付けおよび形態素解析の精度は約 3%低い．しかし，現在は未知語に対して読みと品詞を付与していないので，未知語は誤りに計数されている．読みと品詞を付与する処理を実装すれば，この差は 1%程度になると予想される．

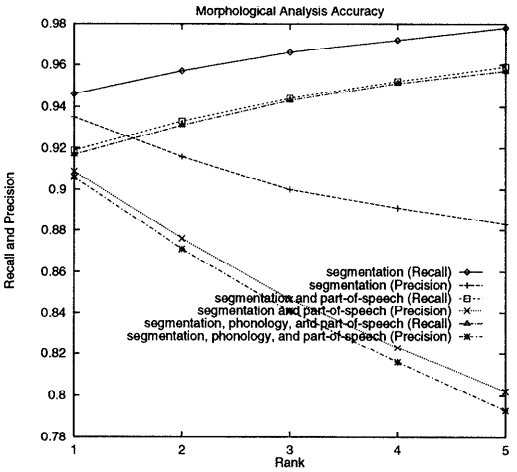

図 6 区切り・読み・品詞が正しい単語の割合
Fig. 6 Word segmentation, pronunciation, and part-of-speech assignment accuracy.

仮名漢字変換やスペルチェッカにおける候補選択のような対話的な確認修正作業では，できるだけ上位の候補に正解が出現することが望ましい．図 6 から分かるように，上位 N 候補の再現率は増加率が逓減する単調増加曲線になっており，この条件を満たしている．

5. 考 察

統計的言語モデルを利用した日本語形態素解析および未知語処理の従来研究には，確率文節文法[15]，隠れマルコフモデルを用いた漢字複合語の自動分割[22]，造語モデル[17]，文字タイプとひらがな bigram を用いた単語分割[24]などがある．

確率文節文法[15]は形式的には日本語の文節構造規則を確率正規文法の生成規則で表現したものであるが，実質的には品詞 bigram 確率と品詞別単語出現確率の積で文の生成確率を近似する言語モデルである．ただし，データ不足の問題を避けるためにすべての品詞別単語出現確率を文字 bigram 確率の積で近似する．小説から抜粋した文を対象とする実験では，文節数最小法よりも誤りが少ないと報告されている．確率文節文法は単語の出現頻度の情報を必要としないので，未知語を自然に扱えるという利点を持つ．しかし，実験で示したように単語出現確率は形態素解析の精度向上に最も貢献する情報なので，本論文の単語分割モデルの方が確率文節文法より精度が高いと予想される．

隠れマルコフモデルを用いた漢字複合語の自動分割法[22]は，接辞と語基を内部状態とし文字を出力シンボルとする隠れマルコフモデルにより，科学技術論文の抄録データに対して約 95%の単語分割精度を得たと報

告されている．この方法はforward-backwardアルゴリズムによりテキストからパラメータを推定する点で，人手によりタグ付けされたコーパスを必要とする本論文の手法よりも優れている．しかし，この手法の適用範囲は漢字複合語に限定されており，広範囲のテキストを対象とする文の形態素解析における有効性には疑問がある．最近の研究では，タグ付きコーパスをまったく使用せずに隠れマルコフモデルのパラメータを推定した場合，新聞記事に対する解析精度は約80%しかないと報告されている[23]．

造語モデル[17]は，文字の表記と読みの組のマルコフモデルを用いて単語（未知語）の生起確率を推定する．単語辞書の見出しを学習データとして，造語モデルが辞書登録語および未登録語に妥当な確率が割り当てるかどうかを調べ，良好な結果を得たと報告している．しかし，この造語モデルを用いた形態素解析の精度は報告されていない．造語モデルは，文字の表記と読みを基本単位とする点では，文字の表記のみを利用する本論文の単語モデルより優れている可能性がある．しかし，造語モデルは単語長分布を考慮していない．実験で示したように単語長分布を考慮する単語モデルの方が単純なマルコフモデルより頑健性は高い．したがって，今後の課題として，単語の表記と読みの両方を考慮した単語モデルを検討する必要があると思われる．

文字タイプとひらがなbigramを用いた単語分割法[24]では，4種類の文字タイプ（漢字，カタカナ，句読点と記号，アルファベットと数字）および83個のひらがなに関するbigram頻度に基づいて，2つの文字の間に単語境界を置くべきかどうかを決定する．この方法は単語辞書を必要としない点で優れているが，漢字列やカタカナ列は対象外である点に問題がある．

文献24)では新聞の社説に対する単語分割の精度を再現率98.3%適合率94.4%と報告されているが，彼らの評価法は非常に楽観的で本論文の方法とはまったく異なる．彼らはシステム出力の単語分割が審判となる人間にとって許容できない場合のみを誤りと計数している．これに対し，本論文ではシステムの単語分割がコーパスの単語分割と一致しないときは無条件に誤りと計数している．

たとえば，図5の形態素解析の例において，システムは「国公立大」という文字列を1つの単語と認定しているが，コーパスの正解は「国公立」と「大」に分割されているので，これは誤りとして計数されている．しかし，少なくとも筆者にとっては「国公立大」という単語分割も許容可能である．このほか図5の例では，「で」（助動詞/助詞）や「全」（名詞/接頭語）の品詞についても議論の余地がある☆．

日本語単語分割の性能評価の問題点は，多くの人が合意できる唯一の正解が存在しないことである．一般に，本論文のように唯一の正解（EDRコーパスの単語分割）との完全一致に基づく評価は，（ある被験者の）許容可能性に基づく評価よりも過小評価になる傾向がある．したがって，異なるシステムの精度を比較する際には，どのような条件で精度の評価が行われたかを十分に吟味する必要がある．本論文の形態素解析法の単語分割の精度は，コーパスの正解との完全一致に基づく評価では再現率94.6%適合率93.5%であるが，許容可能性に基づく評価では再現率99.7%適合率99.5%である．

6. おわりに

本論文では，統計的言語モデルとN-best探索アルゴリズムを用いた日本語形態素解析法について述べた．本論文で提案した日本語形態素解析法の特徴は，人手により単語分割と品詞付与が行われたコーパスから単語分割モデルおよび単語モデルを学習し，確率が大きい順に任意個の形態素解析候補を求めることである．

本手法の今後の課題は，未知語に対して読みと品詞を付与することである．大部分の解析誤りは未知語の周辺で発生するので，読みと品詞を付与できる未知語の確率モデルを研究することは，本手法の完成度を高めるとともに，さらなる精度向上をもたらすと期待される．

参考文献

1) Black, E., Flickenger, A.D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. and Strzalkowski, T.: A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars, *Proc. Speech and Natural Language Workshop*, pp.306–311 (1991).

2) 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).

☆ 図5の例では「全」を名詞とする正解データよりも，接頭語とするシステム出力の方が妥当な解釈であろう．人手で作成した正解データに揺れや誤りが含まれるのは避け難い．しかし，評価の際に正解データの修正を許すと，異なる手法間の厳密な比較が難しくなるので，本論文ではコーパスを絶対的な正解と見なす．

3) 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).
4) Church, K.W.: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, *Proc. 2nd Conference on Applied Natural Language Processing*, pp.136–143 (1988).
5) Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P.: A Practical Part-of-Speech Tagger, *Proc. 3rd Conference on Applied Language Processing*, pp.133–140 (1992).
6) Good, I.J.: The Population Frequencies of Species and the Estimation of Population Parameters, *Biometrika*, Vol.40, pp.237–264 (1953).
7) He, Y.: Extended Viterbi Algorithm for Second Order Hidden Markov Process, *Proc. IEEE 9th International Conference on Pattern Recognition*, pp.718–720 (1988).
8) 久光 徹, 新田義彦：ゆう度付き形態素解析用の汎用アルゴリズムとそれを利用したゆう度基準の比較, 電子情報通信学会論文誌 (D-II), Vol.J77-D-II, No.5, pp.959–969 (1994).
9) Jelinek, F.: Self-Organized Language Modeling for Speech recognition, Technical Report, IBM T.J. Watson Research Center (1985).
10) Katz, S.M.: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer, *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol.ASSP-35, No.3, pp.400–401 (1987).
11) 北 研二, 中村 哲, 永田昌明：音声言語処理—コーパスに基づくアプローチ, 森北出版 (1996).
12) 牧野 寛, 木澤 誠：べた書き文の分かち書きと仮名漢字変換—二文節最長一致法による分かち書き, 情報処理学会論文誌, Vol.20, No.4, pp.337–345 (1979).
13) 松本裕治, 北内 啓, 山下達雄, 平野善隆, 今一修, 今村友明：日本語形態素解析システム「茶筅」version 1.0 使用説明書, 奈良先端科学技術大学院大学 (1997).
14) 松本裕治, 黒橋禎夫, 宇津呂武仁, 妙木 裕, 長尾 真：日本語形態素解析システム JUMAN 使用説明書 version 2.0, 京都大学 (1994).
15) 松延栄治, 日高 達, 吉田 将：確率文節文法による構文解析, 情報処理学会研究報告, 86-NL-56-3, pp.1–8 (1986).
16) Merialdo, B.: Tagging Text with a Probabilistic Model, *Computational Linguistics*, Vol.20, No.2, pp.155–171 (1994).
17) 永井秀利, 日高 達：日本語における単語の造語モデルとその評価, 情報処理学会論文誌, Vol.34, No.9, pp.1944–1955 (1993).
18) Nagata, M.: A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A^* N-Best Search Algorithm, *Proc. 15th International Conference on Computational Linguistics*, pp.201–207 (1994).
19) 日本電子化辞書研究所：EDR 電子化辞書仕様説明書 (第 1 版) (1995).
20) Soong, F.K. and Huang, E.-F.: A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition, *Proc. IEEE International Conference on Acoustic Speech and Signal Processing*, pp.705–708 (1991).
21) Sproat, R., Shih, C., Gale, W. and Chang, N.: A Stochastic Finite-State Word-Segmentation Algorithm for Chinese, *Computational Linguistics*, Vol.22, No.3, pp.377–404 (1996).
22) 武田浩一, 藤崎哲之助：統計的手法による漢字複合語の自動分割, 情報処理学会論文誌, Vol.28, No.9, pp.952–961 (1987).
23) 竹内孔一, 松本裕治：隠れマルコフモデルによる日本語形態素解析のパラメータ推定, 情報処理学会論文誌, Vol.38, No.3, pp.500–509 (1997).
24) Teller, V. and Batchelder, E.O.: A Probabilistic Algorithm for Segmenting Non-Kanji Japanese Strings, *Proc. 12th National Conference on Artificial Intelligence*, pp.742–747 (1994).
25) 吉村賢治, 日高 達, 吉田 将：文節数最小法を用いたべた書き日本語文の形態素解析, 情報処理学会論文誌, Vol.24, No.1, pp.40–46 (1983).
26) 吉村賢治, 武内美津乃, 津田健蔵, 首藤公昭：未登録語を含む日本語文の形態素解析, 情報処理学会論文誌, Vol.30, No.3, pp.294–301 (1989).

(平成 10 年 6 月 26 日受付)
(平成 11 年 7 月 1 日採録)

永田 昌明 (正会員)

1985 年京都大学工学部情報工学科卒業. 1987 年同大学院工学研究科修士課程修了. 同年, 日本電信電話株式会社入社. 1989 年 ATR 自動翻訳電話研究所へ出向. 1993 年日本電信電話株式会社へ復帰. 現在, サイバースペース研究所勤務. 音声翻訳, 統計的自然言語処理の研究に従事. 工学博士. 電子情報通信学会, 人工知能学会, 言語処理学会, ACL 各会員.