自然言語処理 128－2
（1998. 11. 5）

PPM*モデルによる日本語単語分割

小田 裕樹　北 研二

徳島大学 工学部
〒770-8506 徳島市南常三島町 2-1
{hiroki,kita}@is.tokushima-u.ac.jp

日本語処理において，単語の同定，すなわち文の単語分割は，最も基本的かつ重要な処理である．本稿では，文字モデルとビタビアルゴリズムから成る新しい単語分割手法を提案する．まず，本研究の基本法である文字 n-gram モデルに基づく単語分割法を提案する．さらに，単語分割の精度向上のために，文字 n-gram モデルに代わり，データ圧縮アルゴリズム PPM* を用いた言語モデルを適用して単語分割モデルを改良する．PPM* は，確率・統計的テキスト圧縮技法として最も性能の優れた PPM (Prediction by Partial Matching) の一種である．PPM*モデルは無限長文脈を取り扱うことのできる可変長 n-gram モデルとみなすことができる．ATR 対話データベースを用いた評価実験において，文字 n-gram モデルを用いた場合との単語分割精度の比較を行い，本手法の有効性を示す．

Japanese Word Segmentation by a PPM* Model

Hiroki Oda　Kenji Kita

Faculty of Engineering, Tokushima University
2-1, Minami-josanjima, Tokushima, 770-8506 Japan
{hiroki,kita}@is.tokushima-u.ac.jp

Word segmentation, which segments an input sentence into words, is the most fundamental process of Japanese language processing. In this paper, we present a new method for segmenting the input sentence into words, which is suitable for those languages that have no delimiter between words, such as Japanese and Chinese. It consists of a character-based language model and an efficient Viterbi search algorithm. First, we present a word segmentation model based on a character-based n-gram model, which is our basic method. Next, we improve the word segmentation model by using a PPM*-based language model which is a variant of variable length n-gram models, instead of the character-based n-gram model. The effectiveness has been confirmed by evaluation experiments using the ADD (ATR Dialogue Database) corpus.

1　はじめに

自然言語処理において，日本語や中国語等の，単語間に空白を入れる習慣のない言語を対象とする場合，単語分割，すなわち，文を単語列へ分割する処理が必要となる．単語分割は日本語処理の最も基本的で重要な役割を果たす技術であり，精度・速度ともに高い水準の性能が要求される．単語分割と品詞付けから成る形態素解析の最も伝統的な手法は，入力文を単語辞書と照合し，形態素解析候補に順位を与えるために人手により作成された制約や優先規則を用いる方法である．しかし，実際に，すべての単語を辞書で網羅するのは不可能であり，さらに，未知語という重大な問題がある．

一方，英語の品詞付けでは，タグ付きコーパスを用いた確率的手法が確立されている [1, 2, 3]．英語の正書法は単語間で分かち書きするため，これらの手法は，単語モデル (word-based model) を用いている．単語モデルを日本語に適用するためには，いくつかの問題がある．日本語では，未知語の存在が単語の同定に影響を与える上，分割が曖昧で，異なる長さの多くの分割候補があり，それらの候補を比較する．単語モデルを用いるためには，分割候補の確率を正規化する必要が生じる [4]．

最近では，日本語形態素解析にも統計的手法が用いられている [5]．言語モデルとして，文字に品詞情報および文字区切り情報を付加した拡張文字の n-gram モデル [4] や，形態素クラスタリングによるクラス n-gram モデル [6] を用いる研究なども行われている．また，単語モデルによる研究において，可変長 n-gram モデルの一種を用いた英語品詞付け [7] や日本語形態素解析 [8] も提案されている．

本稿では，日本語単語分割の方法として，文字モデル (character-based model) に基づくものを提案する．山本ら [4] が指摘しているように，文字モデルには次のような利点がある．

1. 日本語で，一般的に用いられる文字は約 3,000 ～ 6,000 種類に限られる．これは，単語数に比べるとはるかに少ない．したがって，文字モデルでは，確率モデルのパラメータ数が少なくなり，頑健な推定を行うことができる．

2. 日本語漢字は表意文字であり，一文字が意味を持つ．さらに，平均単語長は約 2～3 文字であるので，一文字は単語に近い情報量を持つと考えられる．

3. 文字モデルでは辞書を使用しない．したがって，未知語の概念そのものがなくなり，未知語の問題を考慮しなくてよい．

4. 一文を構成する文字の数は一定である．したがって，単語分割候補を比較するために確率を正規化する必要がない．

文字モデルとしては，可変長 n-gram モデルの一種である PPM* モデル [9] を用いる．その基礎となる PPM(Prediction by Partial Matching) モデル [10, 11] は，データ圧縮アルゴリズムの一種であり，確率・統計的圧縮技法の中で，Canterbury Corpus を用いた実験で最も優れた性能が示されている．テキスト圧縮は確率的言語モデルの最も直接的な応用であり，圧縮率の良い圧縮法ほど，モデルのエントロピーが小さい．実際に，PPM モデルの自然言語のモデル化への適用も報告されている [12]．PPM* モデルは，PPM モデルを無限長の文脈を取り扱うことができるように拡張したものであり，事前にモデルの次数の上限値を決めることなく，最適な長さの文脈を条件部にもつ確率を推定することができる．PPM* を自然言語のモデル化へ適用することにより，次数に上限のない可変長 n-gram モデルを構築することができる．

以下，本稿では，文字モデルとビタビアルゴリズム (Viterbi algorithm) から成る単語分割手法を提案する．まず，本研究の基本法である文字 n-gram モデルに基づく単語分割モデルを示す．さらに，文字モデルとして，文字 n-gram モデルに代わり，PPM* モデルを用いることによって単語分割モデルを改良する．ADD(ATR Dialogue Database) コーパスを用いた評価実験において，文字 n-gram モデルを用いた場合と，PPM* モデルを用いた場合の単語分割精度を比較し，提案した手法の評価を行う．

2 文字 n-gram モデルに基づく単語分割法

本節では，文字 n-gram モデルに基づく単語分割モデルを提案する．この単語分割モデルが本研究の基本となる．文字 n-gram モデルでは，言語の文字生成は，$(n-1)$ 重マルコフモデルで近似される．長さ l の文字列 $c_1^l = c_1, \cdots, c_l$ において，直前の $(n-1)$ 文字のみが次の文字の生成確率に影響する．実際によく用いられるモデルは，$n=2$ か $n=3$ のモデルであり，それらは，bigram モデル，trigram モデルと呼ばれている．

単語分割モデルの学習データとしては，単語境界位置の付与されたデータを用いる．図 1に学習データの例を示す．記号 (d) は単語境界を表し，(s) と (/s) はそれぞれ文頭と文末の記号である．

単語境界位置の付与された学習データを用いて，文字 n-gram モデルを作成し，これを用いて単語分割を行う．与えられた文を単語列に分割するためには，文中の各文字位置に対し，その文字の前で単語分割が起こるか否かを求めればよい．このために，

⟨s⟩ もしもし ⟨d⟩ 、 ⟨d⟩ 通訳電話国際会議事務局 ⟨d⟩ です ⟨d⟩ か ⟨d⟩ ？ ⟨/s⟩
⟨s⟩ はい ⟨d⟩ 、 ⟨d⟩ そう ⟨d⟩ です ⟨d⟩ 。 ⟨/s⟩
⟨s⟩ 会議 ⟨d⟩ に ⟨d⟩ 申込み ⟨d⟩ たい ⟨d⟩ の ⟨d⟩ です ⟨d⟩ が ⟨d⟩ 。 ⟨/s⟩

図 1: 学習データの例

それぞれの文字位置に対し，2 つの状態 1 と 0 を仮定する．状態 1 と状態 0 は，それぞれ，その文字の前が単語境界となるか否かを表す．文字位置 $i(\geq 2)$ の状態の推定は次式で与えられる．なお，$P_j(c_1^i)$ は文字列 $c_1^i = c_1 \cdots c_i$ を生成して状態 j に到達する確率を表す．

$$P_0(c_1^i) = \max(P_0(c_1^{i-1})A_i,\ P_1(c_1^{i-1})B_i) \quad (1)$$

$$P_1(c_1^i) = \max(P_0(c_1^{i-1})C_i,\ P_1(c_1^{i-1})D_i) \quad (2)$$

$$A_i = p(c_i|c_{i-2}c_{i-1})$$
$$B_i = p(c_i|\langle \mathrm{d} \rangle c_{i-1})$$
$$C_i = p(\langle \mathrm{d} \rangle|c_{i-2}c_{i-1})p(c_i|c_{i-1}\langle \mathrm{d} \rangle)$$
$$D_i = p(\langle \mathrm{d} \rangle|\langle \mathrm{d} \rangle c_{i-1})p(c_i|c_{i-1}\langle \mathrm{d} \rangle)$$

また，$i = 1$ の場合は，次式で求めることができる．

$$P_0(c_1) = p(c_1|\langle \mathrm{s} \rangle) \quad (3)$$

$$P_1(c_1) = 0 \quad (4)$$

ここで，学習データ中の文字位置 1 の前に境界記号がないため，式 (4) を定義する．

文 $s = c_1^m$ に対する最適な単語分割は，各文字位置に対する状態 1 と 0 の最適な状態遷移系列として与えられる．単語分割モデルの計算のため，実際の入力文には，文頭記号と文末記号を各々 0 番目と $m+1$ 番目の文字として加えて処理を行う．学習データ中の文末記号 ⟨/s⟩ の前には単語境界 ⟨d⟩ がないので，最適な状態遷移系列は

$$\max P_0(c_1^{m+1}) \quad (5)$$

となるような状態遷移系列である．これを求めるためには，動的計画法の一種であるビタビアルゴリズム (Viterbi algorithm) を用いることができる (図 2 参照)．

求められた最尤状態遷移系列において，状態 1 である文字位置の前で単語分割を行う．図 2において単語境界を点線で示す．これにより，入力文に対して最適な単語分割を得ることができる．

3 PPM*モデルの適用

文字 n-gram モデルは，文字の生起を $(n-1)$ 重マルコフ過程により近似したモデルであり，文字の生起が過去の $(n-1)$ 文字に依存して決められる．ここで，単純に考えると，n の値を大きくすればより精度の高いモデルが得られると考えられる．しかし，モデルの次数を大きくするにつれ，モデルのパラメータ数が指数的に増大し，必要な学習データの量を著しく増大させる結果となるため，学習データから統計的に信頼性の高いパラメータ値を推定することがますます難しくなる．

そこで，本節では，n-gram モデルの信頼性を損なうことなく，同時に精度を向上させるために，n の値を可変にして，最適な長さの文脈を用いる可変長 n-gram モデル (variable-length n-gram model) を単語分割モデルの言語モデルとして用いる．可変長 n-gram モデルを用いることにより，実際の言語現象での，長い文字列の条件部を必要とする場合や，逆に短い文字列で文字の生起が予測できるものに対応できる．

3.1 PPM*モデル

PPM*[9] は，データ圧縮アルゴリズム PPM[10, 11] を拡張したものであり，確率推定の条件部として無限長文脈を取り扱うことができる．PPM*モデルを自然言語のモデル化へ適用することで，モデルの次数の上限値という制約なしで，可変長 n-gram モデルを実現できる．テキスト圧縮の場合には，テキスト中の一文字を読み込むごとに確率値が更新される動的なモデルとして用いられるが，本研究で用いる PPM*に基づく言語モデルは静的なモデルであり，学習データから獲得した確率値が，評価データによって更新されることはない．以下，基本である PPM モデルの確率推定法と，PPM*モデルの原理を，Cleary ら [9] の例を用いながら簡単に説明する．

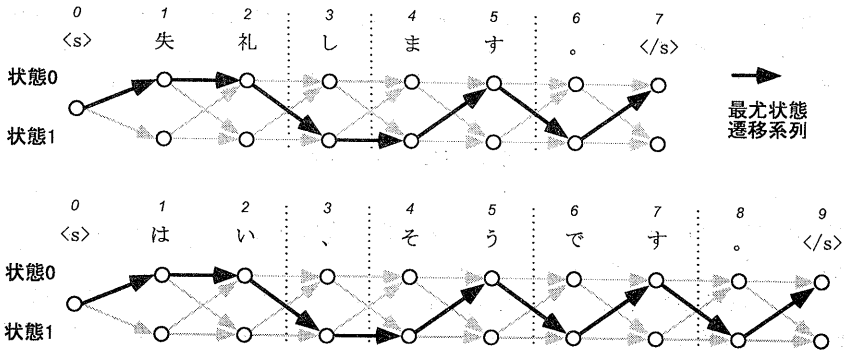

図 2: ビタビアルゴリズムを用いた文の分割

(a) PPM モデルによる確率の推定

文字予測の際に直前の k 文字を用いるモデルを次数 k の有限文脈モデルと呼ぶ．PPM は異なる次数の文脈モデルを組み合わせて文字予測を行う．PPM モデルでは，k の値の最大値は事前に決定しておく．各々の次数のモデルで，学習データ中にすでに出現したすべての長さ k の部分文字列とそれに続いて出現している文字 (予測) の各々の出現回数 c，および，出現回数に基づく予測確率 p を得る．例として，入力文字列 (学習データ) abracadabra を処理した後の $k = 2, 1, 0, -1$ に対する確率値を表 1に示す．

PPM モデルにより確率の推定を行うとき，まず，最高次数 k のモデルを文字予測に用いる．ある文脈が文字を予測できる場合，文字の生起確率としてその予測確率 p を用いる．ただし，次数 k の文脈では予測できない未知文字であった場合，その文脈におけるエスケープ確率を用いて，一つ次数の低い文脈に変更する．エスケープ確率とは各次数の文脈における未知文字に対して用いられる確率であり，未知文字を予測するために，一つ次数の低いモデルへ遷移する確率である．文字を予測できる文脈となるまで，次数 k を小さくする処理を続け，その予測可能な文脈での予測分布に基づいて文字の生起確率を求める．ここで，どのような文字の生起確率を求める場合にも必ず処理が終了するように，文字集合 A 中のすべての文字を等確率 $1/|A|$ で予測する最低次 $k = -1$ のモデルを用意しておく．

以上の処理によって，エスケープ確率の値に依存する割合で，各次数での確率分布を組み合わせて，文字

表 1: PPMC モデル (最高次数 2) の確率値

次数 $k=2$ 予測			c	p
ab	→	r	2	$\frac{2}{3}$
	→	*Esc*	1	$\frac{1}{3}$
ac	→	a	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
ad	→	a	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
br	→	a	2	$\frac{2}{3}$
	→	*Esc*	1	$\frac{1}{3}$
ca	→	d	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
da	→	b	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
ra	→	c	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$

次数 $k=1$ 予測			c	p
a	→	b	2	$\frac{2}{7}$
	→	c	1	$\frac{1}{7}$
	→	d	1	$\frac{1}{7}$
	→	*Esc*	3	$\frac{3}{7}$
b	→	r	2	$\frac{2}{3}$
	→	*Esc*	1	$\frac{1}{3}$
c	→	a	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
d	→	a	1	$\frac{1}{2}$
	→	*Esc*	1	$\frac{1}{2}$
r	→	a	2	$\frac{2}{3}$
	→	*Esc*	1	$\frac{1}{3}$

次数 $k=0$ 予測			c	p
→	a	5	$\frac{5}{16}$
→	b	2	$\frac{2}{16}$
→	c	1	$\frac{1}{16}$
→	d	1	$\frac{1}{16}$
→	r	2	$\frac{2}{16}$
→	*Esc*	5	$\frac{5}{16}$

次数 $k=-1$ 予測			c	p
→	A	1	$\frac{1}{\|A\|}$

を予測する．エスケープ確率の決定には様々な方法が提案されている．標準となっている方法 C (Method C) を用いた PPM を PPMC と呼び，PPM* では，PPM*C が最も良い性能を示している．方法 C は，エスケープ事象の回数として，その時点で文脈に現れている異なり文字の数を与える [11]．本稿では，PPM*C アルゴリズムに基づく言語モデルを単語分割モデルへ適用する．基本的に確率の計算方法は，PPM*C と PPMC とで同じである．以下では，PPMC の確率計算を Cleary ら [9] の例を用いて説

明する．

表1のモデルを用いて文字の生起確率を推定する例を表2に示す．前述したように，文字予測は最高次のモデル（$k=2$）から始める．例において，文字 c が文字列 abracadabra に続いて出現する生起確率を推定する場合，最高次 $k=2$ の文脈 ra から文字 c が確率 1/2 で予測できるため，文字 c の生起確率は 1/2 となる．

表 2: PPMC モデルによる 3 文字の生起確率

文字	予測確率 (排他なし)	予測確率 (排他あり)	生起確率
c	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
d	$\frac{1}{2}$, $\frac{1}{7}$	$\frac{1}{2}$, $\frac{1}{6}$	$\frac{1}{12}$
t	$\frac{1}{2}$, $\frac{3}{7}$, $\frac{5}{16}$, $\frac{1}{\boxed{A}}$	$\frac{1}{2}$, $\frac{3}{6}$, $\frac{5}{12}$, $\frac{1}{\boxed{A}-5}$	$\frac{5}{48} \cdot \frac{1}{251}$

文字列 abracadabra に続いて文字 d が出現する確率を推定する場合，最高次 $k=2$ の文脈 ra からは d を予測できない．そのため，文脈 ra において，予想外の文字に対して割り当てられている確率 1/2 のエスケープ事象が生じ，一つ低い次数 $k=1$ の文脈 a を用いる．これにより，予測 a→d から，文字 d を確率 1/7 で予測できる．したがって，文字 d の生起確率は，確率の積をとって $1/2 \times 1/7 = 1/14$ となる．しかし，ここでさらに，文脈 a には予測 a→c が存在するが，そもそも文字 c の生起確率を求めるのであれば，高次の $k=2$ の文脈 ra から予測されたはずであるので，実際には，この状況で文脈 a から文字 c を予測することはありえないことが分かる．したがって，予測 a→c を考慮しないことで，この文脈における予測確率をより正確に推定することができる．この方法を排他 (exclusion) と呼び，表2に示すように，文脈 a からの文字 d の予測確率を 1/6 へ修正する．このため，最終的に，文字 d の生起確率は 1/12 となる．

学習データ中に出現していない文字 t が，文字列 abracadabra に続いて出現する確率を推定する場合，最高次 $k=2$ から最低次 $k=-1$ までエスケープ事象が繰り返されることになる．次数 $k=-1$ では，高次ですでに現れた文字の確率空間を排他する場合を除き，すべての記号を等確率とする．アルファベット 256 文字の文字集合の場合，t は最低次 $k=-1$ で確率 1/251 で予測され，3 度のエスケープ事象の確率との積をとるため，文字 t の生起確率は $1/2 \times 3/6 \times 5/12 \times 1/251 = 5/12048$ となる．

PPM モデルでは，一般に，最大文脈長を 5 を超える長さとしても性能は向上せず (むしろ悪くなる)，最大文脈長を 5 としたときが，最も良い性能となることが実験的に示されている [10, 11, 13, 14]．これは，PPM が，予測できるまで次数 (文脈長) を減少させるために，エスケープ確率を何回も用いることから，生起確率の推定が準最適の方法で計算されてしまうことを意味している．

(b) PPM* モデルの原理とトライ構造

文脈長に上限値を設ける PPM モデルに対して，PPM* モデルは，文脈長に事前に定める限界値がなく，状況に応じて無限に文脈の長さを変化させることができる．しかし，前述したように，文脈を長くすることは，より多くの情報を得る利点よりも，その情報を活かせない状況が増加する欠点の方が大きい．

そこで，PPM* モデルでは，決定性文脈 (deterministic context) という考えを導入する．文脈が一つの予測のみを与えるとき，その文脈は決定性であると定義する．決定性文脈に対して，未知文字が観測される頻度は一様事前分布に基づく予想よりも低いことが実験的に示されている [15]．

すでに述べたように，PPM モデルでは，文字予測の際に，まず最高次の文脈を予測に用いた．それに対し，PPM* モデルは，文字予測の際に最短の決定性文脈を用いる．もし決定性文脈がない場合は，代わりに最長の文脈を選択する．選択された文脈から，前述した PPMC の計算法を用いることで文字の生起確率を求める．

以上が，PPM*C モデルの文字予測処理である．これにより，常に固定上限値以下の長さの文脈によって予測を行うのではなく，状況に依存した長さの文脈を予測に用いることができる．上の方法では，5 から 6 の長さの文脈が最も多く選択され，さらに，入力が多いほど，それより長い決定性文脈が選択される回数が増えることが報告されている [9]．

次に，PPM* モデルのデータ構造について簡単に説明する．無限長文脈の使用による記憶量と計算時間の問題を解決するために，モデルの格納に用いるトライから入力文字列をポインタで指すようにする (図3参照)．そのポインタを入力ポインタと呼ぶ．特

に，文脈が一意的である (一度しか出現していない) ならばいつでも，トライの葉は入力文字列中を指すことができる．文脈を長くする場合は，入力ポインタを動かすことのみ必要となる．トライを更新するために，現在有効な文脈へのポインタのリスト (先頭は最長文脈) を保持しておく．その結果，作成されるデータ構造を文脈トライと呼ぶ．

例として，文字列 abracadabra に対する文脈トライを図3に示す．入力文字列 (学習データ) 中にすでに出現している文脈は，点線で示す入力ポインタが指す位置が一意になるまで深く伸び，入力文字列中の文脈の先頭へ戻って格納される．たとえば，木の左端部分では，a, ab, abr, abra はどれも一意的でなく，すべて入力文字列中に二回以上出現している．それに対して，abrac は一意的である．したがって，トライの葉 c から入力文字列の先頭 a へ入力ポインタを指す．PPM*モデルの文字予測は，各々の文脈に続く文字の頻度に基づくため，それらの出現回数は，文脈トライの各節に格納され，更新時に頻度をカウントする．

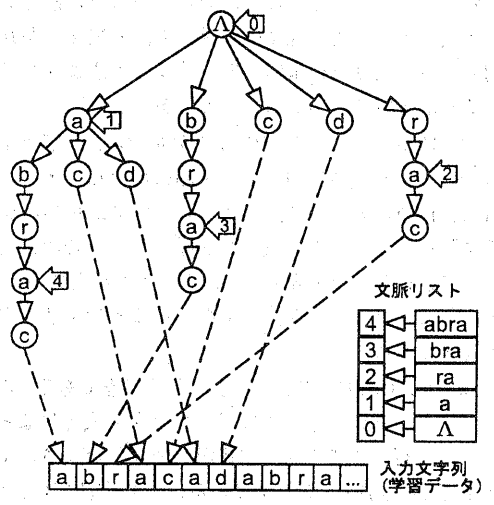

図 3: 文字列 abracadabra に対する文脈トライ

図3中の右側に文脈リストを示す．文脈リストは入力文字列中の現在の位置に関係し，現在有効な文脈へのポインタを含む．これを文脈ポインタと呼ぶ．これらは図中で 0 から 4 で番号付けされ，対応する節を番号付き矢印で示す．最長文脈 abra はリストの先頭へ置かれ，以下の各文脈は順に一文字短くなる．文脈リストの要素数は，最長文脈の長さに (根の) 1 を加えたものである．したがって，リストは常に少なくとも一つの節 (根) を含むことになる．

学習データから構築された文脈トライ[1] を参照することで，PPM*C モデルによる文字の生起確率の推定を行うことができる．たとえば，学習データ abracadabra から構築した文脈トライ (図3参照) を用いて，評価データ中の abracadabra の次に c が出現する確率を求める場合，文字列 abracadabra もしくはその接尾辞の中から最短の決定性文脈を探す．この場合，最短の決定性文脈 ra が発見できるので，文脈 ra から，PPM*C アルゴリズムを用いて文字 c の生起確率を推定する．ただし，決定性文脈が一つも発見できなかった場合は，決定性でない文脈のうち最長のものから，PPM*C アルゴリズムを用いる．

3.2 単語分割モデル

前節で提案した文字 n-gram モデルに基づく単語分割モデルの精度向上を目的として，文字 n-gram モデルに代わり，PPM*モデルを文字モデルとして用いる．単語境界位置の付与された学習データ (図1参照) を用いて，日本語文字や単語境界等の記号を予測単位とする PPM*モデルを作成する．その結果，作成されるトライを参照することで得られる文字の生起確率を用いて，単語分割を行う．

単語分割法は，基本的に前節の文字 n-gram モデルを用いた場合と同じである．ただし，PPM*モデルを適用するために，可変長の文字列を確率の条件部とできるようにする必要がある．なお，単語分割は，一文中での処理であるため，無限長文脈を用いるからといって，条件部の先頭が，文頭記号より前の文字となることはない．前節の単語分割モデルでは，入力文における文字位置 $i \geq 2$ の推定で文字 trigram モデルの確率値を用いており，文字位置 i の前と文字位置 $i-1$ の前に単語境界があるか否かを条件として確率比較を行った．したがって，PPM* モデルを適用するためには，条件部の文字位置 $i-2$ より前における単語境界 (d) の扱いが問題となる．ここで，各文字位置における推定で，文頭まで戻ってすべての文字位置の前で単語境界があるかないかを条件として確率推定を行うことは，文を構成する文字

[1] トライの構築に関する詳細は文献 [9] において説明されている．また，実際には，高速化や記憶容量の節約のために，パトリシアトライのように，分岐のない節を圧縮することや，文脈の接尾辞を指す接尾辞ポインタを追加する [9, 14].

数に比例して，比較する分割候補が指数的に増加するため，実用的でない．

単語分割モデルの式(1)と式(2)におけるパラメータ A_i, B_i, C_i, D_i は，各々，文字位置 i と文字位置 $i-1$ の前に単語境界があるか否かの条件における確率である．ここで，各文字位置の状態(1または0)で確率推定を行うとき，1つ前の文字位置 $i-1$ の状態からは文頭までの最尤状態系列をビタビアルゴリズムで求めることができることを考慮する．これを利用することで，本稿では，PPM*モデルを用いた場合，条件部における文字位置 $i-2$ より前の部分は，最尤状態系列から得られる記号列(単語境界記号を挿入した文字列)とすることで，近似解を求めることとする．

単語分割モデルの基本式は前節同様である．ただし，PPM*モデルを用いた単語分割モデルでは，パラメータ A_i, B_i, C_i, D_i を次式によって推定する．

$$A_i = p(c_i|\ldots c_{i-2}c_{i-1})$$
$$B_i = p(c_i|\ldots c_{i-2}\langle \mathrm{d}\rangle c_{i-1})$$
$$C_i = p(\langle \mathrm{d}\rangle|\ldots c_{i-2}c_{i-1})p(c_i|\ldots c_{i-2}c_{i-1}\langle \mathrm{d}\rangle)$$
$$D_i = p(\langle \mathrm{d}\rangle|\ldots c_{i-2}\langle \mathrm{d}\rangle c_{i-1})p(c_i|\ldots c_{i-2}\langle \mathrm{d}\rangle c_{i-1}\langle \mathrm{d}\rangle)$$

たとえば，A_i を求める場合は，文字位置 i の前と文字位置 $i-1$ の前がともに単語境界がない場合の確率であるので，条件部の接尾辞は $c_{i-2}c_{i-1}$ となる．条件部を2次より大きくする必要がある場合，前述したように，文字 c_{i-2} より前で境界記号を挿入するかどうかを決定する必要がある．この場合は，文字 c_{i-1} の前に単語境界がない条件であるので，1つ前の文字位置 $i-1$ の状態0からビタビアルゴリズムを用いて文頭記号までの最尤状態遷移系列 $\max P_0(c_1^{i-1})$ を求め，文字 c_{i-2} より前の記号列を特定する．つまり，文字 c_{i-2} の前に境界記号があるかどうかも含めて，状態遷移系列 $\max P_0(c_1^{i-1})$ において状態1となる文字の前に境界記号 $\langle \mathrm{d}\rangle$ を挿入した記号列を条件部の接頭辞とする．以上により，決定された条件部の文字列から，学習データから構築されたトライを参照して，まず最短の決定性文脈(ない場合は，最長の文脈)を探し，PPM*C アルゴリズムに基づいて文字 c_i の生起確率を求める．

入力文 $s = c_1^m$ の文末まで以上の処理を行い，文 s に対する最適な状態遷移系列 $\max P_0(c_1^{m+1})$ をビタビアルゴリズムを用いて求めることで，文 s に対して最適な単語分割を求めることができる．

4 評価実験

以上で提案した単語分割の方法を評価するために，ADD(ATR Dialogue Database) コーパスを用いた評価実験を行った．それぞれのデータの文数，単語数，文字数を表3に示す．

表3: クローズドデータとオープンデータのサイズ

	クローズドデータ	オープンデータ
文数	10,000	2,697
単語数	141,658	31,724
文字数	251,699	58,522

単語分割の性能は，再現率(*recall*)と適合率(*precision*)により評価する[5]．ここで，Std をコーパス中の単語数，Sys を本手法で分割された単語数，M を照合した単語数とすると，再現率と適合率は，recall = M/Std，precision = M/Sys で表される．

表4に，PPM*モデルによる単語分割モデルと，文字 n-gram モデル(バックオフ・スムージング[16])による単語分割モデルを適用した場合の単語分割精度を示す．文字 n-gram モデルによる単語分割の精度も決して低いものではないが，PPM*モデルにより単語分割を行った結果のほうが，はるかに精度が良い．単語分割の精度は，オープンデータでさえかなり高精度となっている．

表4: 単語分割モデルの精度

言語モデル	クローズドテスト		オープンテスト	
	再現率	適合率	再現率	適合率
n-gram	97.60%	98.34%	96.79%	97.49%
PPM*	99.70%	99.80%	97.65%	98.24%

本実験においては，PPM*モデルを用いた場合，文字 n-gram モデルを用いた場合と比較して，計算時間が増加した．確率推定時の計算量を減らすための改良を行うことは今後の課題であり，計算時間の比較に関する実験は，改めて行う予定である．

5 おわりに

本稿では，日本語のような単語間で分かち書きをしない言語のための新しい単語分割モデルを提案した．入力文に対して最適な単語分割を見つけるために，本手法は確率的言語モデルとビタビアルゴリズムを用いる．ADD(ATR Dialogue Database) コーパスを用いた評価実験で，文字 n-gram モデルを用いた場合と，PPM*モデルを用いた場合の単語分割精度の比較を行い，PPM*モデルによる単語分割モデルがかなり高精度であるという結果を得た．これにより，本手法の有効性が示されたといえる．

本稿では，PPM*C モデルを用いたが，PPM には様々な改良モデルが提案されている [14]．今後は，可変長 n-gram モデルとして，本稿で用いた PPM*モデルなどの PPM に基づくモデル (たとえば，PPMD モデル) の自然言語のモデル化への適用とその比較評価を行い，その改良や応用に関する研究を行う予定である．

参考文献

[1] Church, K. W.: "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text", ANLP-88, pp. 136–143, 1988.

[2] Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P.: "A Practical Part-of-Speech Tagger", ANLP-92, pp. 133–140, 1992.

[3] Charniak, E., Hendrickson, C., Jacobson, N., and Perkowitz, M.: "Equations for Part-of-Speech Tagging", AAAI-93, pp. 784–789, 1993.

[4] 山本幹雄, 増山正和: "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, 1997.

[5] Nagata, M.: "A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-A^* N-Best Search Algorithm", COLING-94, pp. 201-207, 1994.

[6] 森信介, 長尾眞: "形態素クラスタリングによる形態素解析精度の向上", 自然言語処理, Vol. 5, No. 2, pp. 75–103, 1998.

[7] Schütze, H. and Singer, Y.: "Part-of-Speech Tagging Using a Variable Memory Markov Model" ACL-94, pp. 181–187, 1994.

[8] 春野雅彦, 松本裕治: "文脈木を利用した形態素解析", 情報処理学会研究報告, 96-NL-112, pp.31-36, 1996.

[9] Cleary, J. G. and Teahan, W. J.: "Unbounded length contexts for PPM", *Computer Journal*, Vol. 40, No. 2, pp. 67–75, 1997.

[10] Cleary, J. G. and Witten, I. H.: "Data compression using adaptive coding and partial string matching", *IEEE Transactions on Communications*, Vol. 32, No. 4, pp. 396–402, 1984.

[11] Moffat, A.: "Implementing the PPM data compression scheme", *IEEE Transactions on Communications*, Vol. 38, No. 11, pp. 1917–1921, 1990.

[12] Teahan, W. J. and Cleary, J. G.: "The entropy of English using PPM based models", *Proc. Data Compression Conference*. IEEE Society Press, pp. 53–62, 1996.

[13] Bell, T. C., Cleary, J. G., and Witten, I. H.: *Text compression*, Prentice Hall, NJ, 1990.

[14] Bunton, S.: "Semantically motivated improvements for PPM variants", *Computer Journal*, Vol. 40, No. 2, pp. 76–93, 1997.

[15] Teahan, W. J.: "Modeling English text", D.Phil. thesis, University of Waikato, New Zealand, 1997.

[16] Katz, S. M.: "Estimation of Probabilities from Sparse Data for the Language Model Component of Speech Recognizer", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 3, pp. 400–401, 1987.