自然言語処理 130－1
（1999. 3. 4）

文字クラスモデルに基づく日本語単語分割

小田 裕樹† 森 信介†† 北 研二†

† 徳島大学 工学部
{hiroki,kita}@is.tokushima-u.ac.jp
†† 日本IBM 東京基礎研究所
mori@trl.ibm.co.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，文字クラスモデルを言語モデルとして用いる新しい単語分割手法を提案する．まず，本研究の基本である文字モデルに基づく単語分割法を示す．その上で，文字モデルに代わり，日本語文字のクラス分類により得られた文字クラスモデルに基づく単語分割モデルを提案する．文字クラスモデルでは，推定すべきパラメータ数が文字モデルより少ないという大きな利点があり，文字モデルより頑健な推定を可能とする．ATR対話データベースを用いた評価実験において，文字モデルを用いた場合と文字クラスモデルを用いた場合の単語分割精度の比較を行い，文字クラスモデルに基づく本手法の有効性を示す．

Japanese Word Segmentation Based on a Character Class Model

Hiroki Oda† Shinsuke Mori†† Kenji Kita†

† Faculty of Engineering, Tokushima University
{hiroki,kita}@is.tokushima-u.ac.jp
†† Tokyo Research Laboratory, IBM Japan
mori@trl.ibm.co.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 is based on a character class model. First, we explain a word segmentation model using a character-based model. Next, we present a word segmentation model based on a character class model instead of the character-based model. The effectiveness has been confirmed by evaluation experiments using the ADD (ATR Dialogue Database) corpus.

1 はじめに

日本語や中国語等においては，単語間に空白を入れる習慣がないため，これらの言語の計算機処理では，まず文を単語列に分割する処理が必要となる．単語分割は日本語処理における最も基本的かつ重要な技術であり，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日本語形態素解析法の多くは，単語辞書の登録語との照合を行い，複数の形態素解析候補がある場合は人手により作成した規則を用いて候補間の順位付けを行うというものである．しかし，実際に，辞書中にすべての

単語を網羅するのは不可能であるため，未知語 (辞書未登録語) という重大な問題が生ずる．

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]．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 [2]．

以上の点から，我々は文字モデルに基づく単語分割法を提案した [3, 4]．文字モデルは，未知語モデルとしても機能するために，学習データに含まれていない単語に対してもある程度の対応が可能である．さらに，本稿では，より頑健な単語分割モデルを構築するために，日本語文字のクラスタリング (グループ化) を行うことを考える．日本語漢字は表意文字であり，一文字が意味を持つ．したがって，何らかの基準によりいくつかのグループ (クラス) に分類することが可能である．文献 [2] で示されている文字モデルの利点に加え，文字クラスモデルでは，文字モデルよりもさらにモデルのパラメータ数を減らすことができるという大きな利点がある．したがって，文字クラスモデルは文字モデルよりも頑健な推定を可能とするので，単語分割へ適用した場合，未知語に対する頑健性がさらに向上すると考えられる．

文字とクラスの対応関係を得るためのクラスタリング処理には，削除補間法を拡張した平均クロス・エントロピーを言語モデルの評価基準としたクラスタリング法 [5] を用いる．平均クロス・エントロピーを評価基準として求められた単語 bigram クラスモデルは，単語 bigram モデルよりも予測力という点において優れていることが実験的に示されている [5, 6]．本稿では，このアルゴリズムを日本語文字のクラスタリングに適用し，文字クラスモデルを構築する．

以下，本稿では，文字クラスモデルに基づく新しい単語分割手法を提案する．まず，基本となる文字モデルに基づく単語分割モデルを簡単に説明する．さらに，類似した文字を自動的にグループ化するクラス分類法について説明し，文字クラスモデルに基づいた単語分割モデルを提案する．ADD(ATR Dialogue Database) コーパスを用いた評価実験において，文字モデルを用いた場合と，文字クラス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 文字モデルに基づく単語分割法

本節では，文字モデルによる単語分割手法 [3, 4] について説明する．まず，言語モデルとして，文字 trigram モデルを用いることを考える．文字 trigram モデルでは，言語の文字生成は，2 重マルコフモデルで近似される．したがって，文字列において，直前の 2 文字のみが次の文字の生成確率に影響する．

単語分割モデルの学習データは「⟨s⟩ は い ⟨d⟩ 、⟨d⟩ そ う ⟨d⟩ で す ⟨d⟩ 。 ⟨/s⟩」のように単語境界を付与したデータを用いる．ここで，特殊記号 ⟨d⟩, ⟨s⟩, ⟨/s⟩ は各々「単語境界」,「文頭」,「文末」を表している．

単語境界位置の付与された学習データから文字 trigram モデルの確率値を計算し，これを用いて単語分割を行う．与えられた「ベタ書き」文を単語列に分割するためには，入力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，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．

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

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

$$A_i = p(c_i|c_{i-2}c_{i-1})$$

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

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

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

ここで，文字位置 1 の状態 0 の確率は，$P_0(c_1) = p(c_1|\langle \mathrm{s} \rangle)$ により求めることができる．また，学習データ中の文字位置 1 の前には境界記号がないため，状態 1 の確率 $P_1(c_1)$ は 0 と定義する．

入力文 $s = c_1^m$ 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 $\max P_0(c_1^{m+1})$ 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・アルゴリズムを用いることができる (図 1参照)．

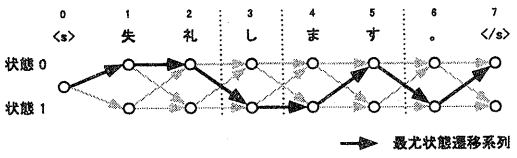

図 1: ビタビ・アルゴリズムを用いた文の分割

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 1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．

また，同様の考えに基づいて可変長 n-gram モデルを用いた単語分割を行うことも可能である [3, 4]．その場合は，解探索における単語分割候補の指数的増加を避けるために，各文字位置において確率の高い候補のみを後続する文字位置での探索に用いるようにする．もし文字 trigram モデルによる単語分割モデルと同様に，文字位置 i の直前が単語境界である (1) か否 (0) かの 2 つの仮定に対する各々の最尤解のみに関して解探索を行うならば，その探索空間は，図 1 に示す探索空間と同じとなる．

3 日本語文字のクラスタリング

3.1 文字 n-gram クラスモデル

n-gram モデルに，クラスという概念を導入したモデルを n-gram クラスモデル (n-gram class model) と呼ぶ．ここで，クラスとは n-gram モデルの予測単位とする文字 (あるいは単語) の集合を何らかの基準で分類あるいはクラスタリングしたものを指す．本節では，特に日本語漢字が表意文字であり，一文字が意味を持っていることから，類似した文字を自動的にグループ化することを考える．

文字クラス数は文字数に比べると少ないものとなるので，文字 n-gram モデルよりも文字 n-gram クラスモデルの方が推定すべきパラメータ数が少ないという利点がある．また，文字クラスモデルは，文字クラスを用いた一種のスムージングであり，頑健なモデルな構築することが期待できる．このため，文字 n-gram クラスモデルは，文字 n-gram モデルよりも必要な学習データ量が少なく，たとえ小さな学習データからでも，信頼性のある確率値を推定することが比較的容易となる．

文字 n-gram クラスモデルでは，次の文字を直接予測するのではなく，先行する文字クラス列から次の文字クラスを予測した上で次の文字を予測する．ここで，文字が一つのクラスにしか属さないとすると，文字の生起確率は次の式で表すことができる．

$$P(c_i|c_1^{i-1}) = P(c_i|\mathcal{C}_i)P(\mathcal{C}_i|c_{i-n+1}^{i-1}) \qquad (3)$$

クラス $\mathcal{C}_i$ は，文字 c_i の属する文字クラスである．また，確率 $P(c_i|\mathcal{C}_i)$ は次式により最尤推定できる．

$$P(c_i|\mathcal{C}_i) = \frac{N(c_i)}{N(\mathcal{C}_i)} \qquad (4)$$

ここで，$N(c_i)$ は学習データ中で文字 c_i が出現した回数であり，$N(\mathcal{C}_i)$ はクラス $\mathcal{C}_i$ の文字が出現した回数である．

さらに，本稿では，未知文字を考慮するために，未知文字のクラスを考える．未知文字クラスには，学習データ中に出現しない未知文字と，頻度の小さい文字を含めることとする (未知文字の実例の収集)．未知文字 c が未知文字クラス $\mathcal{C}$ から生起する確率 $P(c|\mathcal{C})$ は，対象言語の文字集合 A と，既知文字集合 A_k から，$1/|A - A_k|$ と計算できる．

3.2 文字クラスタリング法

クラス分類法には様々なものが提案されている [7]．たとえば，ある程度パラメータの数が少なくなるまで，学習データのエントロピーを小さくするように文字をグループ化する方法が考えられる．しかし，当然であるが，学習データに対して最適なクラス分類を行うことが，対象言語に対しての最適解であるとは限らないため，得られた文字クラスモデルが文字モデルより優れた言語モデルである保障はない．実際に，言語モデルは，未知の評価データを用いて評価されるので，理想的には，対象言語の未知のデータに対してエントロピーを小さくするように文字をグループ化することが望ましいと考えられる．

文献 [5] では，以上の点から，学習データ内の一部を未知の評価データとして扱い，その評価データ

のエントロピーが小さくなるようにクラス分類を行うアルゴリズムを提案している．このアルゴリズムには，停止基準を評価基準から導き出せるという利点があり，停止条件が人間の判断に委ねられることはない．実際に，得られた単語 bigram クラスモデルは単語 bigram モデルよりも優れた性能を誇ることが実験的に報告されている．そこで，本稿では日本語文字のクラスタリングに文献 [5] の手法を適用することを考える．

3.2.1 クラスタリングの評価基準

クラスタリングの評価基準として用いる平均クロス・エントロピーについて説明する．ここで，言語モデルの性能尺度であるクロス・エントロピー H は以下の式で定義される．

$$H(M,T) = -\frac{\sum_{i=1}^{n} \log p_M(s_i)}{\sum_{i=1}^{n} |s_i|} \qquad (5)$$

ここで，M は言語モデル，s_i は評価データ T 中の i 番目の文である．$|s_i|$ は文 s_i を構成する文字の数となる．このとき，文区切りを考慮するために，s_i は文末記号までを含むと仮定する．

学習データ内に未知の評価用データを用意して，その評価データによりクラス分類の性能を評価する．これを実現するために，削除補間と同じ技術を用いる．

1. 学習データ L を m 個の部分データ $L_1, ..., L_m$ に分割する．
2. 各部分データ $(i = 1, ..., m)$ に対し，ステップ 3, 4 を行う．
3. 学習データから L_i を削除し，残りの $m-1$ 個のデータから確率値を推定する．
4. 削除されたデータ L_i で，式 (5) によりクロス・エントロピーの値を計算する．

以上のようにして，m 個のクロス・エントロピーの値を得ることができるので，それらの値の平均値 $\overline{H}$ (平均クロス・エントロピー) を全体の評価関数とする．

$$\overline{H} = \frac{1}{m} \sum_{i=1}^{m} H(M_i, L_i) \qquad (6)$$

ここで，M_i はステップ 3 で L_i を削除した残りのデータから推定されたモデルである．この平均クロス・エントロピーの値を最小とする文字とクラスの対応関係を求めることが，本稿の文字クラスタリングの最終目的となる．

3.2.2 クラスタリング・アルゴリズム

文字クラスモデルを構築するためには，文字クラスタリングにより文字とクラスの対応関係を求めることが必要となる．文字とクラスの対応関係としては，ある文字が一定の確率で複数のクラスに属するという確率的な関係も考えられるが，解空間が広大になるので，本稿では，文字は一つのクラスのにみ属することを仮定する．

以下では，文字とクラスの対応関係を返すクラス関数 f を用いて説明する．たとえば，文字 c_1 の属するクラスとして，$f(c_1) = \{c_1, c_2, c_3\}$ を返す．このとき，文字 c_2, c_3 に対するクラス関数 f も，各々の文字が属するクラスとして同じく文字集合 $\{c_1, c_2, c_3\}$ を返すこととなる．ここで，クラスタリング対象文字の集合を A_k とすると，A_k 中のすべての文字のクラス関数 f の和集合は A_k となり，A_k と未知文字クラスの和集合が対象言語の文字集合 A となる．

さらに，文字のクラス分類に対する解探索を行うために，文字とクラスの対応関係の変更を表す関数 $move$ を定義する．移動関数 $move$ は，文字とクラスの関係 f に対して，文字 c をクラス $\mathcal{C}$ に移動した結果得られる文字とクラスの関係を返す．文字は唯一のクラスに属するとしているので，$move(f, c, \mathcal{C})$ は，現在，文字 c が属するクラス f から，集合の要素 c を取り除き，クラス $\mathcal{C}$ に要素 c を加えることを意味する．

文字クラス分類の最適解を求めるためには，あらゆる可能な文字とクラスの対応関係を調べる必要がある．クラス分けの総数は有限であるので，理論的には総当たり戦略により最適なクラスを見つけることはできる．しかし，総当たり法は非現実的であるため，準最適なアルゴリズムを用いることとなる．文献 [5] のアルゴリズムを以下に示す．

文字クラスの学習アルゴリズム

文字集合 A_k 中の文字を頻度の降順にソートし，$c_1, \cdots, c_n$ とする

foreach $i(1, 2, \cdots, n)$

$\mathcal{C}_i := \{c_i\}$

$f(c_i) := \mathcal{C}_i$

foreach $i(2,3,\cdots,n)$
  $\mathcal{C} := \mathbf{argmin}_{\mathcal{C}\in\{c_1,c_2,\ldots,c_{i-1}\}} \overline{H}(move(f,c_i,\mathcal{C}))$
  if $(\overline{H}(move(f,c_i,\mathcal{C})) < \overline{H}(f))$ **then**
    $f := move(f,c_i,\mathcal{C})$

上記アルゴリズムは，初期状態において，各文字を各々一つのクラスとみなしている．後は，頻度の高い文字の順に他のクラスへの文字の移動を仮定して，平均クロス・エントロピーの値を再計算している．このとき，平均クロス・エントロピーが減少する文字とクラスの新しい対応関係が発見できれば，クラス関数 f を変更する．頻度の高い文字から処理を行う理由は，頻繁に出現する文字ほどクロス・エントロピーに与える影響が大きいと考えられるので，早い段階での移動が後の移動によって影響されにくく，収束がより速くなると考えられるからである．クラスタリングの処理の例を図2に示す．

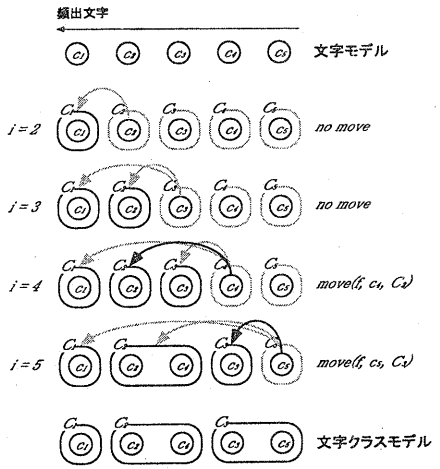

図 2: 文字クラスタリングの処理の例

4 文字クラスモデルに基づく単語分割法

文字クラスモデルを言語モデルとして，単語分割を行う．ここで，3節の文字クラスタリング法では，文字と文字クラスの関係が一意に定まることを考えると，一文を構成する文字列 c_1^m がそのまま文字クラス列 $\mathcal{C}_1^m$ に変換できることが分かる．単語分割モデルでは，入力文の各文字間において単語境界の有無を仮定して文の生成確率を計算・比較する．ここで，式(4)から分かるように，確率 $p(c_i|\mathcal{C}_i)$ は単語境界の有無には影響を受けない値である．さらに，一文を構成する文字は不変であるので，$\prod_{i=1}^{m} p(c_i|\mathcal{C}_i)$ はどのような分割候補の確率を求める場合でも一定の値の項となる(式(3)参照)．したがって，文字 trigram クラスモデルによる単語分割モデルは，以下のようにクラス連鎖の確率のみを用いて簡単に計算することができる．

$$P_0(c_1^i) = \max(P_0(c_1^{i-1})A_i,\ P_1(c_1^{i-1})B_i) \quad (7)$$

$$P_1(c_1^i) = \max(P_0(c_1^{i-1})C_i,\ P_1(c_1^{i-1})D_i) \quad (8)$$

$$A_i = p(\mathcal{C}_i|\mathcal{C}_{i-2}\mathcal{C}_{i-1})$$
$$B_i = p(\mathcal{C}_i|\langle \mathrm{d}\rangle\mathcal{C}_{i-1})$$
$$C_i = p(\langle \mathrm{d}\rangle|\mathcal{C}_{i-2}\mathcal{C}_{i-1})p(\mathcal{C}_i|\mathcal{C}_{i-1}\langle \mathrm{d}\rangle)$$
$$D_i = p(\langle \mathrm{d}\rangle|\langle \mathrm{d}\rangle\mathcal{C}_{i-1})p(\mathcal{C}_i|\mathcal{C}_{i-1}\langle \mathrm{d}\rangle)$$

ここで，文字位置 1 の状態 0 の確率は $P_0(c_1) = p(\mathcal{C}_1|\langle \mathrm{s}\rangle)$ となり，状態 1 の確率は $P_1(c_1) = 0$ となる．

上記の単語分割モデルをみれば分かるように，文字クラスモデルを用いた場合は，文字クラスの連鎖により単語境界を予測するという問題に置き換わる．文字 trigram クラスモデルを用いた場合も，$\max P_0(c_1^{m+1})$ となる状態遷移系列をビタビ・アルゴリズムを用いて求めることで，入力文に対する最適な単語分割を得ることができる(図1参照)．また，可変長 n-gram クラスモデルを用いる場合でも，同様に，クラス連鎖における単語境界の出現の有無により確率比較を行い，解探索を行うこととなる．

5 評価実験

以上で提案した手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1に示す．

5.1 文字クラスモデルの性能評価

まず，前節の単語分割モデルで用いる文字クラスモデルを作成する．したがって，文字 trigram クラスモデルや可変長 n-gram クラスモデルの予測力を

表 1: 学習データとテストデータのサイズ

	学習データ	テストデータ
文数	11,430	1,267
単語数	155,553	17,829
文字数	278,771	31,450

改善するような文字クラスを求める必要がある．しかし，文字クラスタリングアルゴリズムの評価基準である平均クロス・エントロピーの計算を考えると，高次のモデルでは，必要な記憶容量と計算時間が大きな問題となる．

そこで，本実験では，クロス・エントロピーの計算は，低次の bigram 確率によって計算した．bigram モデルであれば，高速なクラスタリング処理が可能である．もし日本語における文字分類の最適解に近い解を得ることができれば，得られたクラス関数 f はどのような次数のモデルに対してもある程度有効であると考えられる．

また，本稿では，日本語文字が明らかに字種によって分類できることから，クラスタリング処理において，字種により規制を設けることを考えた．たとえば，漢字は漢字同士でグループ化するというように考えることで，文字とクラスの対応関係の変更を考える場合に必要な計算量を少なくすることができる．これにより，漢字の場合は *move* 関数の移動先クラスとして漢字のクラスのみを考えることとなり，ひらがなの場合はひらがなのクラスのみとなる．

以上の条件により，文字クラスタリングを行うために，学習データを 9 個のデータ $L_1, ..., L_9$ に分割した．ここで，1 個のデータにしか出現しない文字は未知文字とし[1]，字種ごとに未知文字クラスを用意した．したがって，クラスタリングの対象となる文字は，2 個以上のデータに出現する文字となる．また，単語分割に用いる言語モデルを獲得することを念頭におくため，単語間に単語境界記号を挿入した分かち書きデータを用いた．単語境界記号自体はクラスタリングの対象ではないが，その存在により，エントロピー評価では単語境界 (単語間のスペース) まで考慮するようになる．

本実験において，評価データ中の未知文字 (クラスタリング対象文字以外) は字種ごとに異なる特別な記号に置き換えてクロス・エントロピーの計算を行った．未知文字の扱いは文字モデルと文字クラスモデルで共通であるので，未知文字の確率はモデルの比較においては問題とならない．重要なことは，クラスタリング対象文字のグループ化によって，モデルの予測力がどのように変化するかである．以上の点から，モデルの状態は，既知文字すべて (もしくは文字クラスすべて)，未知文字クラス (字種ごと)，単語境界，文区切りの各々に対応することとなる．実験により得られた，文字 bigram モデルと文字 bigram クラスモデルのクロス・エントロピーを表 2に示す．

表 2: 言語モデルのクロス・エントロピー

言語モデル	H
文字 bigram モデル	3.5980
文字 bigram クラスモデル	3.5591

本実験において，文字クラスモデルのクロス・エントロピーは文字モデルのものよりも小さく，より予測力の高い言語モデルの獲得に成功している[2]．また，表 3に，クラスタリング対象文字数とそれらをクラスタリングした後の文字クラス数を示す．学習データ中には，1357 種類の文字が含まれていたが，約 200 種類の低頻度文字が未知文字として取り扱われた．実験の結果，クラス当たりの平均要素 (文字) 数は 1.36 文字であり，最大のクラスの所属文字数は 12 文字であった．

文字クラスタリング実験により得られたクラス関数 f が返す文字集合 (文字クラス) を，図 3にいくつか示す．必ずしもすべての文字クラスが言語直観から納得がいくものではないが，いくつかのノイズと思われる文字を除けば，ある程度良い解が得られていることが分かる．不自然な印象を受ける文字のグループが存在するのは，あくまで bigram クラスモデルの改善における準最適解を求めていることを原因としていると考えられる．

本実験では，各文字クラスに属する文字数は少な

[1] 平均クロス・エントロピーの計算における各々の未知文字をそのまま学習データ全体における未知文字の収集に用いている．

[2] 単語間のスペースを考慮しない「ベタ書き」の日本語データを用いた実験では，クロス・エントロピーは，文字 bigram モデルでは 4.3563 ビット，文字 bigram クラスモデルでは 4.3060 ビットであり，同様に，文字クラスモデルのほうが一文字当たりのエントロピーが小さいという結果が得られている．

表 3: 文字数と文字クラス数の比較

	既知文字数	文字クラス数
ひらがな	70	67
カタカナ	78	71
漢字	935	675
英字	42	13
数字	10	8
記号	23	15
合計	1158	849

{ 送, 乗, 居, 貼 } { 私, 誠, 娘, 又 } { 思, 誓 }
{ 今, 昨 } { 部, 型 } { 中, 低 } { 他, 皆, 僕 }
{ 原, 松, 草 } { 山, 竹, 塚, 吉 } { 別, 誰 } { 特, 既 }
{ 忙, 楽 } { 近, 多, 暗 } { 渡, 貸, 探, 押 } { 朝, 昼 }
{ 市, 職, 命, 履 } { 安, 幸 } { 図, 計, 義 }
{ 島, 木, 川, 根 } { 食, 刻, 飯 } { 帰, 困 } { 女, 性 }
{ 校, 化, 枠, 郊 } { 映, 厳, 撮 } { 含, 休, 混 }
{ 購, 納 } { 離, 訪 } { 項, 故 } { 戸, 宮 }
{ 欄, 横, 机, 縦, 層, 逆 } { 界, 株, 財 }

図 3: 実験により得られた文字クラスの例

いものであり，文字クラスタリングによって，それほど極端にパラメータ数が減少するということにはならなかった．これは，今回用いたコーパスの規模が小さいものであり，学習データに含まれる文字の種類が少なかったことを原因としていると考えられる．今後は，より多くの文字種を含んだ大規模コーパスでの文字クラスタリング実験を試みることを予定している．

5.2 単語分割精度

文字クラスタリング実験により得られたクラス関数 f を用いることで，文字 trigram クラスモデルや可変長 n-gram クラスモデルを構築することができる．ここで，文字クラスタリングでは字種別にグループ化を行ったので，単語分割に用いる文字クラスモデルを作成するときに，クラス関数 f を用いる字種を限定してみることについても試みることとした．もしあまり有効でない文字のグループ化が行われている字種があれば，それらの文字はクラス関数 f を用いず，文字を予測単位として処理すれば，より性能の良いモデルが得られる可能性がある．

また，本稿で提案した文字クラスモデルに基づく単語分割モデルは非常に簡単な構造となっており，いかにクラス連鎖により単語境界の生起を把握するかが単語分割精度の鍵となる．ここで，字種変化によるヒューリスティクスを考慮した場合，片仮名，数字，英字はその字種同士の文字間では分かち書きされる可能性がほとんどないと考えられる．単語分割をする場合，これらの文字は単に片仮名か数字か英字であるという情報のみでモデル化したほうが良い結果が得られる可能性がある．そこで，それらの字種に関しては，字種全体を一つのクラス (グループ) とみなすことについても検討することとした．

以上の点から，文字クラスモデルと文字モデルの比較において，表 4の 5 つのモデルを考え，単語分割実験を行った．表中には，字種ごとに何を予測単位としてモデル化を行うかを示している．(1) は文字モデルであり，(2) は文字クラスタリングの結果に何も手を加えずに，すべての文字でクラス関数 f を用いた文字クラスモデルである．(3), (4), (5) は字種クラス (字種全体を一つのクラスとする) を予測単位とすることを試みたモデルであり，それらの中の (4) と (5) では文字クラスタリングの結果得られるクラス関数 f を用いる文字を限定している．

表 4右に，文字 trigram クラスモデルまたは文字 trigram モデルに基づく単語分割モデルによる単語分割精度を示す[3]．本実験では，バックオフ・スムージング付きの trigram 確率値を計算した．実験結果 (1) と (2) は文字 trigram モデルと文字 trigram クラスモデルの精度であるが，オープンデータにおいて文字クラスモデルの精度が上回る結果となっている．また，(1) と (3) および (2) と (4) の結果を比較することで，片仮名，数字，英字を各々一つのクラスとしたほうがオープンデータに対して，精度が向上していることが分かる．したがって，字種単位でのグループ化の有効な字種の存在が確認できた．全体として，オープンデータでは，漢字に関して文字クラスを用いた (5) の場合が最も高精度であった．

また，可変長 n-gram モデル [4] と可変長 n-gram クラスモデルの比較に関する単語分割実験も行った

[3] 単語分割の性能は，再現率 (*recall*) と適合率 (*precision*) により評価する [8]．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: 文字 trigram モデルまたは文字 trigram クラスモデルによる単語分割精度

	(字種ごとに予測単位を使い分けることを仮定する)						クローズドテスト		オープンテスト	
	漢字	ひらがな	カタカナ	数字	英字	記号	再現率	適合率	再現率	適合率
(1)	文字	文字	文字	文字	文字	文字	98.10%	98.56%	95.48%	94.11%
(2)	クラス	クラス	クラス	クラス	クラス	クラス	98.09%	98.56%	95.83%	94.34%
(3)	文字	文字	字種	字種	字種	文字	97.82%	98.44%	95.91%	94.73%
(4)	クラス	クラス	字種	字種	字種	クラス	97.80%	98.43%	95.86%	94.70%
(5)	クラス	文字	字種	字種	字種	文字	97.83%	98.44%	96.01%	94.74%

が，trigram 同様，クラスモデルの方が高精度であった．本実験において，可変長 n-gram (クラス) モデルによる探索空間は trigram による場合と同じものとした．可変長 n-gram クラスモデルを用いて単語分割を行った場合，最も精度の高い (5) のモデルでは，オープンテストで再現率 96.38%，適合率 96.23% を達成した．したがって，文字クラスモデルは未知語を含むデータに対して文字モデルよりも頑健であることが結論できる．

本実験の結果より，文字クラスタリングの動機であった漢字のクラスタリングにはかなり良い解が得られていることが分かる．また，文字クラスモデルでは，モデルのパラメータ数が少ないために今回用いたような比較的小規模の学習データからでも信頼のおける確率値を得ることが容易となっている．本実験により，日本語文字のクラスタリングにより得られた文字クラスモデルを用いる本手法の有効性を示すことができたといえる．

6 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，本手法は文字クラスモデルを言語モデルとして用いる．ADD コーパスを用いた評価実験で，文字モデルを用いた場合と，文字クラスモデルを用いた場合の単語分割精度の比較を行い，文字クラスモデルによる単語分割モデルが高精度であるという結果を得た．文字クラスタリングにより，言語モデルのパラメータ数を減少させることができているので，より頑健なモデルを得ることができた．これにより，本手法の有効性が示されたといえる．

参考文献

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

[2] 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.

[3] 小田裕樹, 北研二: "PPM* モデルによる日本語単語分割", 情報処理学会研究報告, 98-NL-128, pp. 9–16, 1998.

[4] 小田裕樹, 北研二: "決定性文脈を用いた可変長 n-gram モデルによる日本語単語分割", 言語処理学会第 5 回年次大会, 1999. (発表予定)

[5] 森信介, 西村雅史, 伊東伸泰: "クラスに基づく言語モデルのための単語クラスタリング", 情報処理学会論文誌, Vol. 38, No. 11, pp. 2200–2208, 1997.

[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] Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C. and Mercer, R. L.: "Class-Based n-gram Models of Natural Language", *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, 1992.

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