人名辞書から名前読み付与規則を抽出するアルゴリズム

増田恵子[†] 梅村恭司[††]

日本人は外国語のカタカナ表記を知っていても，その綴りを知らないことが多い．そこで，アルファベット表記とカタカナ表記が対応する規則があると便利である．一方，外国人名のアルファベット表記とカタカナ表記が対応したデータベース（人名辞書）が存在し，これは利用可能である．本稿では，人名辞書におけるアルファベット表記の部分文字列とカタカナ表記の部分文字列の組の出現頻度の変化から，アルファベット表記とカタカナ表記の対応規則を自動的に得るアルゴリズムを提案する．本アルゴリズムは，ローマ字綴りからなる仮想の対応規則を組み合わせたデータ集合から，規則を完全に再現することができる．人名辞書にアルゴリズムを適用して得られた対応規則をデータ検索システムで使われることを想定して評価した結果，正当率は80%，綴りの復元率は84%，読みの復元率は48.8%の精度を得た．また，人手によって規則を作る方法と比較した結果，本手法は人の知識を使用しない方法にもかかわらず人間の生成する規則を含む規則を得ることができた．

An Algorithm that Extracts Alphabet-Kana Rules from Name Database

KEIKO MASUDA[†] and KYOJI UMEMURA[††]

It is often the case that Japanese people knows the words in Kana spelling but does not know its true spelling. It is useful if there are correspondence rules between alphabet and Kana spelling. In this paper, we propose an algorithm to extract automatically Alphabet-Kana correspondence rules from the pairs of alphabet of person names and their corresponding Kana spelling. We have checked the algorithm by the test rules that consisted of Romaji and its Kana spelling, and applied the algorithm to the data composed by this rules. We have confirmed that all of the test rules reappeared in extracting rules. We have applied the algorithm to the actual data and measured the correctness of the extracted rules, considering information retrieval application. As the result, we have obtained 80% as correctness of the rules, 84% as coverage of the rules and 48.8% as correctness of its replacement. Even if our method does not use human knowledge, we are able to get replacement rules which human generates.

1. はじめに

日本人は外国語を発音する場合，日本語の音韻体系の許容範囲内で発音してしまうことが多い[1)]．そのためアルファベット綴りにある音情報は損なわれてしまう．カタカナ表記は，外国語を日本語の音で置き換えたものである．日本人は外国語のカタカナ表記を知っていても，その綴りを知らないことが多い．綴りを知らず，カタカナ表記しか知らない場合，外国語のデータを日本人が検索することは簡単ではない．英語の外来語ならば辞書を参照する方法があるが，人名などの場合には，読まれうるカタカナの表記が一意ではないため，そのような辞書を用意することは実際的でない．そこで，アルファベット表記とカタカナ表記が対応した規則が必要となる．ここで，規則を手で作り出すのはコストがかかるし，規則は古くなるという問題がある．これは，自動的に規則を生成したい要求があることを意味する．また，人名などでは多くの流儀の綴りが混在するために，特定の言語に依存しない方法が望まれる．このような背景から，本稿では，外国語のアルファベット表記とカタカナ表記が対応したデータベースから，対応規則を獲得する方法を提案する．

今までに，辞書検索においてカタカナ表記を使った英単語の検索も行われている[2)]．しかし，これらの方法はアルファベット表記の発音記号を用いている．また，アルファベット表記からカタカナ表記を得るための方法がいくつか提案されている．たとえば，アルファベッ

† 日本電信電話株式会社名古屋支店
Nippon Telegraph and Telephone, Nagoya Branch
†† 豊橋技術科学大学情報工学系
Department of Information and Computer Sciences, Toyohashi University of Technology

ト表記とカタカナ表記の対応規則を，アルファベット表記の発音記号から生成する[3]という方法である．これらは，対象の言語に依存した知識を使えないという前提条件と異なる．

アルファベット表記の発音記号の情報を用いずにカタカナ読みを得る方法もある[4]．しかし読みへの対応において手で行う部分があり，コストがかかるという問題があった．また，英語固有名詞の部分文字列とカタカナ変換テーブルから対応するカタカナを導くという手法[5]においても，変換テーブルを手で作り出す必要がある．これらは，自動的に規則を得るという要求を満たしていない．

本稿では，外国人名のアルファベット表記とカタカナ表記が対応したデータベース（人名辞書）から，アルファベット表記とカタカナ表記の対応規則を自動的に得るアルゴリズムのなかで，理想的なデータを入力すると漏れなく規則が再現できることを特徴とするアルゴリズムを提案する．さらに，このアルゴリズムを記述するばかりでなく，その正当性と評価結果を示す．このアルゴリズムは，データベースにおける文字列の出現頻度の情報を用いて対応規則を抽出する．そして，対応規則の抽出において，アルファベット表記の言語の母音を示す文字を指定する以上の知識は使わない．

本稿の構成は以下のとおりである．2章では，対応規則を抽出するアイデアについて説明する．3章では，対応規則抽出のアルゴリズムを説明し，アルゴリズムで用いるパラメータの実験値を考察する．4章では，人名辞書から対応規則を抽出して得られた規則を示す．5章では，抽出した対応規則の評価方法を述べ，評価実験を行う．また，他の手法との結果の比較を行う．6章で考察を行い，7章でまとめる．

2. 問題定義

アルファベット綴りとそれに対応する日本語読みを1組としたデータの集合から，綴りに対応する日本語読みの規則を得ることが本研究の目的である．そのときに，規則の正確さを損なわない範囲で，できるだけ多くの規則を抽出することに重きを置く．これ以降，綴りに対応する日本語読みの規則を「対応規則」と呼ぶ．

アルファベットの綴りとそれに対応する日本語の読みは，そのものが対応規則の1つであるが，それが多数存在するときには，綴りの一部と読みの一部を対応規則としてよいと人間は推定することができる．たとえば，“nyane”と「ニャネ」があるとき，“nya...”のような単語について，それに対応する読みが「ニャ...」であることが多いならば，人間は“nya”は「ニャ」と読むと推定する．そして，“ne”は「ネ」と読むと推定する．このようにして，規則をあらかじめ用意しておくことで，未知のカタカナ表記のデータについても，検索の対象となりうるアルファベット綴りの集合を求めることができる．

求める規則は，アルファベットを対応するカタカナに変換するために使うのではなくて，カタカナで示された単語に関して，検索の対象となりうる範囲を定めるものである．検索の効率を極端に低下させないためには，その規則のもっともらしさはある水準以上であることが必要である．また，もれなく検索するためには，規則はできるだけ多く求めたい．

本研究での問題は，アルファベット綴りとそれに対応するカタカナ表記の組の集合から，そのような対応規則を求めるものである．これは，その集合の要素の長さ n 文字のアルファベット綴りとそれに対応する長さ m のカタカナ表記の $(n-1)(m-1)$ 個の部分文字列の対について，対応規則としての信頼度を推定し，対応規則となりうるものをすべて求めることをし，これをすべての要素について行うという問題となる．

2.1 記号の定義

これ以降に用いる記号を定義する．データ集合を D で表し，$D=\{d_1,\cdots,d_l\}$，l はデータ数を表す．d_h はデータで，$d_h=(A_h,K_h)$ と表す．(A_h,K_h) はアルファベット文字列 A_h とカタカナ文字列 K_h の組で，$A_h=a_1\cdots a_m$，$K_h=k_1\cdots k_n$ と表す．a_i $(1\leq i\leq m)$ はアルファベット文字で，m は文字数，k_j $(1\leq j\leq n)$ はカタカナ文字で，n は文字数を表す．

$d_h=(A_h,K_h)$ の A_h を語頭と語尾の2つに分ける．同様に，K_h も2つに分ける．2つに分けた A_h と K_h の部分文字列を語頭と語尾でそれぞれ対応させ，部分文字列の組を作る．この部分文字列の組は対応規則となるかもしれない文字列の組，つまり対応規則の候補である．d_h の語頭対応規則候補を $cfak_h^{ij}$，語尾対応規則候補を $crak_h^{ij}$ で表し，その要素を以下に示す．

$$cfak_h^{ij}=(a_1\cdots a_i,\ k_1\cdots k_j)$$
$$crak_h^{ij}=(a_{(i+1)}\cdots a_m,k_{(j+1)}\cdots k_n)$$
$$(1\leq i\leq m-1,\ 1\leq j\leq n-1)$$

（ただし $a_i\cdots a_i$ は a_i と同じ，$k_j\cdots k_j$ は k_j と同じ）

$cfak_h^{ij}$ の集合を $CFAK_h$，$crak_h^{ij}$ の集合を $CRAK_h$ と表す．$CFAK_h$ と $CRAK_h$ はともに $(m-1)(n-1)$ 個の対応規則候補を持つ．

データ集合 D の任意の d_h から作られる $CFAK_h$ と $CRAK_h$ の対応規則候補が D において出現す

表1 出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$
Table 1 Frequency table $F(CFAK_h)$ and $F(CRAK_h)$.

$F(CFAK_h)$

	k_1	$\cdots$	k_{n-1}
a_1	$f(cfak_h^{11})$	$\cdots$	$f(cfak_h^{1(n-1)})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_{m-1}	$f(cfak_h^{(m-1)1})$	$\cdots$	$f(cfak_h^{(m-1)(n-1)})$

a_{m-1} : $a_1 \cdots a_{m-1}$ k_{n-1} : $k_1 \cdots k_{n-1}$

$F(CRAK_h)$

	k_2	$\cdots$	k_n
a_2	$f(crak_h^{11})$	$\cdots$	$f(crak_h^{1(n-1)})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_m	$f(crak_h^{(m-1)1})$	$\cdots$	$f(crak_h^{(m-1)(n-1)})$

a_2 : $a_2 \cdots a_m$ k_2 : $k_2 \cdots k_n$

る頻度を調べる．d_h の語頭対応規則候補 $cfak_h^{ij}$ の D における出現頻度を $f(cfak_h^{ij})$，d_h の語尾対応規則候補 $crak_h^{ij}$ の D における出現頻度を $f(crak_h^{ij})$ と表す．$f(cfak_h^{ij})$ を i, j に関して表としたものを $F(CFAK_h)$ と表し，$f(crak_h^{ij})$ を同様に表としたものを $F(CRAK_h)$ と表す．$F(CFAK_h)$ と $F(CRAK_h)$ を出現頻度表と呼ぶ．**表 1** に $F(CFAK_h)$ と $F(CRAK_h)$ を示す．

3. 対応規則の抽出方法

実験データ集合 D の d_h $(1 \leq h \leq l)$ について，語頭対応規則候補 $cfak_h^{ij}$ と語尾対応規則候補 $crak_h^{ij}$ $(1 \leq i \leq m-1, 1 \leq j \leq n-1)$ を作成し，出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$ を求める方法を示した．この出現頻度表の頻度情報から対応規則を抽出するアルゴリズムを示す．

3.1 前　　提

本研究では，外国語に依存した知識を使わずに対応規則を得たいと考えている．扱うデータは外国人名とその日本語の読みが対になったデータである．外国人名には英語表記やドイツ語表記など複数の言語が含まれるが，アルファベット表記に対するカタカナ表記は必ず日本語である．そこで，日本語の知識は使用することにした．用いたのは次の2つである．

(1) 母音はカタカナ表記の区切りである．
日本語の発音において母音は音の区切りとなる．そこで，アルファベット表記の “a”, “e”, “i”, “o”, “u” は母音であるという情報をアルゴリズムに与える（変音記号のついた母音（ü など）も母音要素とする）．

(2) 促音（ッ）と長音（ー）は語頭に現れない．
たとえば，カタカナ表記「キャット」は「キャッ」と「ト」の2文字として取り扱う．

3.2 アルゴリズム

アルゴリズムで用いるパラメータは次の2つである．

- 信頼限度 (R)：信頼にいたる出現頻度の下限値
- 閾値 (T)：変化点における出現頻度の比の上限値

出現頻度表 $F(CFAK_h)$ の頻度情報から語頭対応規則と語尾対応規則を抽出するアルゴリズムを以下に示す．

```
1:for (j = 1; j < n − 1; j++)
2:   for (i = 1; i < m − 1; i++) {
3:     if (R > f(cfak_h^ij)) return;
4:     if (a_i は母音でない) continue;
5:     if (T > (f(cfak_h^(i+1)j)/f(cfak_h^ij))) {
6:       cfak_h^ij を語頭対応規則として抽出；
7:       crak_h^(i+1)(j+1) を語尾対応規則として抽出；
8:       break;
9:     }
10: }
```

出現頻度表 $F(CRAK_h)$ から語頭対応規則と語尾対応規則を抽出するアルゴリズムを以下に示す．母音に関する知識の処理が上記と異なるが，判定条件は同じである．

```
1:for (j = n − 1; j > 1; j−−)
2:   for (i = m − 1; i > 1; i−−) {
3:     if (R > f(crak_h^ij)) return;
4:     if (a_i は母音である) continue;
5:     if (T > (f(crak_h^(i−1)j)/f(crak_h^ij))) {
6:       crak_h^ij を語尾対応規則として抽出；
7:       cfak_h^(i−1)(j−1) を語頭対応規則として抽出；
8:       break;
9:     }
10: }
```

人名データの実験で用いたパラメータの値は，信頼限度 $R = 10$，閾値 $T = 1/3$ である．これらの値は，対象データのサンプル調査と，日本語の言語モデルから得られる実験値から定めた値である．詳しくは次章で述べる．

アルゴリズムで注意した点を次に述べる．アルゴリズムの1行目で終了条件が $j \leq n-1$, $j \geq 2$ でない理由は，カタカナ表記が1音しかないデータがあるからである．通常人名はアルファベット表記よりカタカナ表記の方が短い．そこでカタカナ文字列を定めて，それに対するもっともらしいアルファベット文字列を求めることにした．これが，1行目で j ループが外で

ある理由である．アルゴリズムの6，7行目において，対応規則の抽出の際に語頭対応規則と語尾対応規則をともに抽出する．この理由は，注目する文字列対の語頭あるいは語尾が対応規則であるならば，その残りの語尾あるいは語頭も対応していると見なせるからである．ともに抽出した語頭対応規則と語尾対応規則は対であるので，対の情報も記録する．この対を語頭・語尾対応規則対と呼び，規則の評価において使用する．

3.3 パラメータの実験値

閾値 仮想的なデータのモデルとして，「外国語のアルファベット表記はアルファベットの前後関係によらない文字列からなる」というモデルを考える．つまり「子音の次には母音がくる」かどうかは独立事象とする．このモデルのデータ集合から，出現頻度表を作成する．この出現頻度表の数の並びは，$F(CFAK_h)$ では $f(cfak_h^{ij})$ は i, j について単調に減少し，$F(CRAK_h)$ では $f(crak_h^{ij})$ は i, j について単調に増加するだろう．しかし，実際のデータにおける出現頻度表の数の並びは単調にならない．注目するカタカナ文字列に対してアルファベット文字列が実際に対応する可能性がなくなったとき，その文字列対の出現頻度は上記に仮定したモデルにおける頻度より急激に減少する．この急激に変化する点を決めるのが閾値である．

信頼限度 使用するデータにはノイズが存在することが多いが，ノイズをすべて取り除くのは一般に困難である．そこで，データにノイズが存在したままで処理を行うが，ノイズの乗った場合，頻度の低いデータは統計的に不安定となるデータなので対応規則を得る過程から除かなければならない．信頼限度は出現頻度数を設定するパラメータである．信頼限度以上の出現頻度数ならば，注目しているデータは信頼できると考える．

閾値の実験値 本研究の目的は，外国語に対応する日本語表記を得ることである．日本語のローマ字綴り，たとえば (a, ア) は対応規則と考えるのが自然である．ゆえに，日本語のローマ字綴りを仮想の対応規則としてモデルを構築することができる．日本語の五十音において母音数が少ないのは「ヤ行」の3つである．「ヤ行」を対応規則で表すと (ya, ヤ)，(yu, ユ)，(yo, ヨ) となる．対応規則のアルファベット表記より，“y” の次が “a” か “u” か “o” である確率は，それぞれ 1/3 であることが分かる．したがって，閾値は最低 1/3 にしなければ，間違えた場所でアルファベット文字列が区切られてしまう可能性がある．この閾値の値は日本語に依存した値である．閾値が 1/3 でよいことを確認するために次の実験を行った．

表2 TAK_{17} から作成した3段合成データ集合での出現頻度表の1つ

Table 2 Frequency Table Sample in CD_3 of TAK_{17}.

$F(CFAK_h)$

	ア	ヤ
a	289	17
y	51	17
a	17	17
n	9	9

$F(CRAK_h)$

	ヤ	ノ
y	17	34
a	17	68
n	17	289
o	17	289

日本語のローマ字綴りからなる対応規則を仮想対応規則とする．仮想対応規則の集合を TAK_q で表す．q は仮想対応規則数を示す．仮想対応規則を組み合わせて，n 段合成データ集合 CD_n を作る．n は組合せの回数を示す．合成データ数は q^n 個となる．n 段合成データ集合 CD_n から出現頻度表を作成しアルゴリズムを適用した結果，抽出した対応規則が仮想対応規則集合 TAK_q のすべての仮想対応規則を再現できれば用いたパラメータは正しい，という仮定で実験を行う．この条件ではデータにノイズは存在しないので，信頼限度は仮に1でよいとする．作成した仮想対応規則の集合 TAK_{17} を以下に示す．作成した仮想対応規則は1対1対応ではなく，アルファベット文字が重なるように定義した．

$TAK_{17} = \{$
(a, ア), (i, イ), (u, ウ), (e, エ), (o, オ),
(na, ナ), (ni, ニ), (nu, ヌ), (ne, ネ), (no, ノ),
(ya, ヤ), (yu, ユ), (yo, ヨ),
(nya, ニャ), (nyu, ニュ), (nyo, ニョ),
(n, ン) $\}$

TAK_{17} の仮想対応規則を総当たりで3回組み合わせて，3段合成データ集合 CD_3 を作成した．合成データ数は $17^3 = 4913$ 個である．TAK_{17} の CD_3 において出現頻度表を作成した．例として d_h = (ayano, アヤノ) の出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$ を**表2**に示す．

この例からは，$F(CFAK_h)$ から対応規則 (a, ア), (yano, ヤノ) が，$F(CRAK_h)$ から対応規則 (aya, アヤ), (no, ノ) が抽出された．TAK_{17} の CD_3 から得られた対応規則は744個あり，この中に TAK_{17} の仮想対応規則17個すべてが含まれていた．したがって，日本語における閾値は 1/3 でよいとする．他言語に対する閾値は同様に求めればよい．

信頼限度の決定法 信頼限度は，信頼できる出現頻度数を設定するパラメータである．出現頻度表での出現頻度が低いということは，その出現頻度を持つ対応規則候補は偶然の一致であって，アルファベット文字列とカタカナ文字列が実際には対応していないケースが

あると考えられる．データが多くなればなるほど，偶然と考えられる頻度は多くなる．また，データのなかにある読みの揺れや，データ作成の過程によっても，偶然と考えられる頻度は変化する．したがって，信頼度に関してはデータのサンプル調査から決定する．

アルゴリズムでは，頻度の比を求めており，信頼限度は注目している対応規則候補の出現頻度を調べるパラメータだが，アルゴリズムにおいて頻度の傾きを調べる際に，次に注目する対応規則候補の出現頻度も調べる．したがって，次に注目する対応規則候補の頻度が，偶然の変動があっても判定が変わらないようにしなければならない．

このことを正確に判定するには，サンプルにおける頻度が小さいときに，これが真の値よりも小さい事象を確率モデルを作って検定するべきであるが，これをサンプル調査することは難しいため，偶然の規則が出現する頻度から推定することにした．

実験に使ったサンプルを調査したところでは，2回繰り返して現れたものが，偶然で現れた対応かどうか判断の分かれ目であった．安全をとり，使用するデータでは3回までは偶然の出現と考えることとする．このことは，頻度3は偶然の出現であるか断定できないということであるが，頻度0と頻度3の区別が信頼できないことを意味すると解釈することにした．頻度5と頻度0が出現すると，アルゴリズムは規則検出するが，頻度5と頻度3では，規則を検出しない．これは不都合である．

頻度の傾き ($slope$) は次のように定義される．$slope = f(cfak_h^{(i+1)j})/f(cfak_h^{ij})$．これより，注目している対応規則候補の出現頻度は $f(cfak_h^{ij}) = f(cfak_h^{(i+1)j})/slope$ で求まる．閾値の実験値は 1/3 なので $slope = 1/3$，$f(cfak_h^{(i+1)j}) = 3$ を代入して，$f(cfak_h^{ij}) = 3/(1/3) = 9$．したがって，実験のデータに関する信頼限度は 10 とする．

一般には，データをサンプル調査し，偶然に出現しうるペアの出現頻度が N であったとき，閾値 $*N$ を超える整数を信頼度限度とするのが妥当である．

3.4 理想データによる検証

閾値の実験値を求める際に，仮想対応規則集合 TAK_{17} から作成した3段合成データ集合 CD_3 にアルゴリズムを適用した．本節では，2段合成データ集合 CD_2 にアルゴリズムを適用し，仮想対応規則の再現を調べる．

TAK_{17} から CD_2 を作成し，その出現頻度表にアルゴリズムを適用して対応規則を抽出した．この結果，TAK_{17} の CD_2 から抽出された対応規則は，TAK_{17} の仮想対応規則のみであることを確認した．

表3 TAK_{183} から作成した2段合成データ集合での出現頻度表
Table 3 Frequency table in data set CD_2 of TAK_{183}.

$F(CFAK_h)$

	ニャ
n	183
y	183
a	183
n	10

$F(CRAK_h)$

	ネ
y	14
a	37
n	183
e	183

また，もう1つの仮想対応規則集合を用意した．この仮想対応規則集合は，日本語の音（カタカナ表記）151音に対するローマ字綴り（アルファベット表記）の組からなり，このうち日本語の32音は2種類のローマ字綴りがある．したがって，仮想対応規則数 $q = 183$ である．この仮想対応規則集合 TAK_{183} から2段合成データ集合 CD_2 を作成し，出現頻度表を作成した．

TAK_{183} の d_h = (nyane, ニャネ) の出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$ を**表3**に示す．表3の出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$ から対応規則 (nya, ニャ)，(ne, ネ) が抽出される．TAK_{183} の CD_2 から抽出された対応規則は，TAK_{183} の仮想対応規則のみであった．

2段合成データ集合から抽出した対応規則は仮想対応規則を完全に再現することができた．したがって，本アルゴリズムは入力のデータが理想的なときに，合成された対応関係から合成前の規則を推定できるアルゴリズムになっていることが分かる．

4. 対応規則の抽出

対応規則の抽出に用いる実験データ[6]は，西洋人名のアルファベット表記とカタカナ表記の対からなるデータの集合であり，使用言語は複数ある．そして，姓と名を分離して，それぞれを用いる．実験データ集合には，アルファベット表記とカタカナ表記が対応しないノイズデータが含まれる．実験データ集合に含まれるおおよそのノイズの割合は，実験データ集合から無作為抽出した100個のデータに含まれるノイズの割合で与える．実験データを，対応規則を抽出するための学習データと対応規則を抽出しない適用データに分ける．適用データ集合は，実験データ集合から100個を無作為抽出したデータの集合，学習データ集合は，それ以外のデータの集合とする．

学習データ集合 Mix は言語混合のデータ集合で，データ数は31847，ノイズの割合は約1%である．適用データ集合 Mix_{ap} にノイズはない．Mix の全データから出現頻度表を作成した．d_h = (limerick, リマリック)

表4 Mix における d_h = (limerick, リマリック) の出現頻度表
Table 4 Frequency table of d_h = (limerick, リマリック).

$F(CFAK_h)$

	リ	マ	リッ
l	347	5	1
i	253	5	1
m	10	5	1
e	1	1	1
r	1	1	1
i	1	1	1
c	1	1	1

$F(CRAK_h)$

	マ	リッ	ク
i	1	1	1
m	1	1	1
e	1	10	11
r	1	39	42
i	1	52	141
c	1	54	427
k	1	65	959

表5 Mix から抽出した語頭・語尾対応規則対の一部
Table 5 Samples of R_f from data set Mix.

{front, rear}
{(brei, ブレ), (t, イト)}
{(korole, コロレ), (v, フ)}
{(lalibe, ラリベ), (la, ラ)}
{(ne, ネ), (her, エル)}
{(reuchli, ルクラ), (n, ン)}
{(scho, ショレ), (ley, イ)}
{(slidel, スライデ), (l, ル)}

の出現頻度表 $F(CFAK_h)$ と $F(CRAK_h)$ を**表4**に示す．表4の出現頻度表より，$F(CFAK_h)$ から {(li, リ), (merick, マリック)} が，$F(CRAK_h)$ から {(limeri, リマリッ), (ck, ク)} と {(lime, リマ), (rick, リック)} が抽出される．

本アルゴリズムによって頻度に基づき抽出された対応規則の集合を R_f と表す．Mix から抽出した語頭・語尾対応規則対の一部を**表5**に示す．Mix から得られる語頭・語尾対応規則対の抽出数は65317，語頭対応規則の抽出数は21770，語尾対応規則の抽出数は16712であった．

5. 対応規則の評価

本アルゴリズムは語頭と語尾の対応規則を抽出するときに，データのアルファベット表記とカタカナ表記をお互いが対応し合うように2つに分けた．この規則をデータベースの検索に利用することに効果があるかを評価するために，次の対応規則を用意した．用意した対応規則は，実験データ集合のアルファベット表記とカタカナ表記の各文字列をそれぞれ半分ずつに分けて，対応をとった規則である．半分に分けるということは，2つに分けることの基準となるので，この規則は抽出した規則の基準となる規則である．半分に分ける際に，アルゴリズムで使用した前提条件の「母音は音の区切りである」ことと，「促音と長音は語頭に現れない」ことも考慮した．この規則をナイーブな対応規則と呼び，その集合を R_n と表す．

表6 Mix から得られるナイーブな語頭・語尾対応規則対の一部
Table 6 Samples of R_n from data set Mix.

{front, rear}
{(banza, バンザ), (rov, ロフ)}
{(ennec, エンネク), (cerus, ツェルス)}
{(hawa, ハワ), (tmeh, トメ)}
{(ja, ヤン), (nca, カ)}
{(lak, ラク), (hdar, ダル)}
{(pax, パクス), (ton, トン)}
{(sal, サラ), (laba, バ)}

ナイーブな対応規則は，その単純さゆえに個々の規則としての自然さは提案方式よりも劣ると考えられるが，多くの種類の可能な対応規則を生成する方法である．データベースの検索では，ある程度の自然さがあれば，多様な規則が生成される方法は有望な方法である．データベースの検索の問題では規則の多様性は好ましい性質であり，ナイーブな方法は，簡単だが有望な方法である．

提案方法とナイーブな方法を同一の人名情報から規則を生成して評価した．使用した人名は，文献6)より，読みとスペルの情報のみを転記したものであり，人名といっても姓，名それぞれ別に分け，その後にアルゴリズムを使用した．このため，2分割をするアルゴリズムではあるが，姓と名に分割する実験ではない．

Mix から得られるナイーブな語頭・語尾対応規則対の一部を**表6**に示す．Mix から得られるナイーブな語頭・語尾対応規則対の抽出数は31736，語頭対応規則の抽出数は11696，語尾対応規則の抽出数は8608であった．一般に対応規則は，以下に述べる正当率，綴りの復元率，読みの復元率から評価する．

正当率 生成された規則が，どのくらい確からしいかが，最初に検討すべき項目である．そのため，対応規則のアルファベット文字列とカタカナ文字列の組が実際の規則として受け入れられるかを調査する．対応規則は d_h を2分割した語頭・語尾対応規則対を持つので，対応規則の正当率を調べるには，d_h のアルファベット表記とカタカナ表記が正しく2つに分割されたかを調べればよい．ここでの「正しい」は人間が判断した結果である．もともと対応がある頻度以上あるものばかりであるので，抽出した結果はそのような関係が存在しうるという意味ではすべて正しいとする考え方もあるが，これは人間が抽出する規則と一致するとは限らない．この正当率が高ければ，人間の考える規則を正解としたときに，余分な規則を生成していないことを意味する．作業としては，語頭・語尾対応規則対の集合から100個無作為抽出したデータを調べ，正しく2分割されていた語頭・語尾対応規則対の割合を

正当率とする．

綴りの復元率 十分な量の規則が生成されているかが，次に検討すべき項目である．このため，対応規則がどれくらい外国人名の綴りをカタカナに変換できるか調べる．変換は以下のように行う．まず，アルファベット表記の先頭から規則を適用する．適用する規則は長い表記のものを優先する．この条件で，適用データのアルファベット表記に対応規則のアルファベット表記が一致するとき，適用データのアルファベット表記を対応規則のカタカナ表記で置き換える．

データベースの検索においては，置換えは効率上問題となるところであるので，複雑なバックトラックを想定するのは実際の状況と異なる．したがって，置換えは最長一致を優先にした単純な方法で行う．また，語頭から置き換える．置換えが失敗したとしても，対応規則を適用する場所や，置換えの場所を変更して再度，置換えをすることはしない条件で置き換える．手で規則を生成するときは，長い規則が短い規則と矛盾がないようにし，この方針でも検索に問題が起きないようにするのが普通である．

この方針の結果，適用データ集合の各アルファベット表記が対応規則集合のカタカナ表記ですべて置き換えられた数を，対応規則の綴りの復元率とする．この条件であるので，置き換えられた読みがもとのデータと違っていても正解と判定する．

読みの復元率 実際のデータベースの検索における検索可能な確率が規則の性能の目安となる．これを計測するために，綴りの復元率の測定と同様に変換し，適用データのアルファベット表記すべてを復元できたデータに対して，置き換えたカタカナ表記の読みが「正しい」かを調べる．「正しい」の判断は，規則の選択肢のなかに，カタカナ表記が正確に出現するかで行う．少しの表記の違い（例：ディとデ）も間違いとする．

綴りを復元できたデータの集合において，読みが正しかった割合を読みの復元率と定義する．この比率は，規則が生成できた場合に，原データにある正解のカタカナ表記で目的のデータが発見できる確率に相当する．「正しい」の判定例を以下に示す．例中の“|”は「または」を表す．

- 正しいとした例
  辞書データ (anselmo, アンセルモ)
  復元データ (ansel, **アンセル**),(mo, **モ** | リモ)
- 間違いとした例
  辞書データ (batmunkh, バトムンフ)
  復元データ (bat, **バト**),(mun, マン | マンゼ | ミュン | **ムン**),(kh, ク | ハ)

表 7 対応規則の評価
Table 7 Comparison of R_f and R_n.

	R_f	R_n
precision (%)	80	68
A-rate (%)	84	98
K-rate (%)	48.8	36.7

Mix での抽出した対応規則 R_f とナイーブな対応規則 R_n の正当率 (precision)，読みの復元率 (K-rate)，綴りの復元率 (A-rate) を**表 7** に示す．表 7 の結果より，対応規則の正当率は，ナイーブな対応規則より抽出した対応規則の方が良かった．また，綴りの復元率は，ナイーブな対応規則の方が良かった．これは，ナイーブな対応規則では，*Mix* の全データから規則を得られるのに対し，抽出する対応規則では，まれな綴りの規則を得ることができないことが原因である．したがって，ナイーブな規則は予想どおり多様な読みに対応し，なんらかのカタカナでデータを検索することはできる．しかし，綴りを復元したデータに対する読みの復元率は，抽出した対応規則の方が良かった．これは，実際のデータベースの検索における検索可能性では提案するアルゴリズムはナイーブなアルゴリズムより効果があることを意味する．これらのことから，提案するアルゴリズムは有効であるといえる．

5.1 言語別データ集合

学習データ集合 *Mix* は言語混合のデータ集合であった．本節では，言語別のデータ集合において上記の評価を行う．これは言語ごとの傾向を見るためであり，データを言語で分けるのは非常に自然であると考える．人名辞書のデータを出身ごとに分類し，言語別のデータ集合を作成した．出身で分類したので，言語別とはいっても他言語も混ざっている．しかしこれについては考慮していない．アメリカあるいはイギリス出身を英語データ集合 *Eng*，フランス出身をフランス語データ集合 *Frn*，ドイツ出身をドイツ語データ集合 *Ger* とした．それぞれのデータ数は 2400 である．またデータ集合のおおよそのノイズ量は，*Eng* で 2%，*Frn* で 1%，*Ger* で 1%である．言語別データの適用データ集合（データ数各 100）にノイズはない．言語別データ集合から得られた対応規則の評価結果を**表 8** に示す．

表 8 の結果より，言語別データ集合での抽出した対応規則の正当率は平均すると 83%で，言語混合データ集合での 80%より良い．言語別データ集合での対応規則の綴りの復元率は平均 47.3%で，言語混合データ集合での 84%の半分ほどしかない．しかし，言語別データ集合のデータ数は 2400 であるのに対し，言語混合

表8 言語別データ集合における対応規則の評価
Table 8 Comparison of *Eng*, *Frn* and *Ger*.

	Eng	*Frn*	*Ger*
precision	86.0	85.0	78.0
A-rate	51.0	48.0	43.0
K-rate	39.2	62.5	55.8

表9 外来語における対応規則の評価
Table 9 Evaluation Result for Loanwords.

	R_f	R_n
precision (%)	79	61
A-rate (%)	84	94
K-rate (%)	50.0	38.3

データ集合はデータ数31847で，かなりの差がある．このことから，綴りの復元率はデータ数にも依存すると思われる．言語別データ集合での読みの復元率は平均52.5%で，言語混合データ集合での48.8%よりも良かった．

読みの復元率から判断すると，対応規則を抽出するためのデータは言語別データの方が良い傾向を示すというデータも得られた．また，綴りの復元率から判断すると，実験に利用したデータ量では，まだ，規則を完全に再現できておらず，データ数を増やす必要があることも判定できる．

5.2 外来語についての実験

本稿のアルゴリズムは，もととなる対応データは望むだけ入手できるという人名を念頭において考案されているが，同じアルゴリズムが辞書にある外来語に適用した結果を参考の情報として示す．外来語は国語辞書[7]より，1単語に相当する外来語を抜き出したもので行った．

外来語は6171語と少ないが，辞書から抜き出したものであるのでサンプル調査しても誤りがないのは明らかである．よって，信頼限度を4として実験した．評価の方法は人名の場合と同様である．**表9**に結果を示すが，表7とほぼ同様な結果が得られた．これは，アルゴリズムが人の名前だけでなく，固有名詞一般に適用できる可能性を示唆している．

5.3 人手による規則生成との比較

データが十分あった場合について，対応規則が十分生成されるが，そのなかに人間が生成するような規則が含まれているかは明らかではない．つまり，本方式で生成した規則で検索して，人間が自然に読みを付与できるような情報について検索洩れが生じないか調べたい．そこで，読みの復元率について，「英語固有名詞の片カナ変換」[5]との結果の比較を試みた．この方式は英語固有名詞をカタカナ変換テーブルを用いてカタカナ読みに変換するもので，カタカナ変換テーブルは手で作り出している．カタカナ読みへの変換は，英語文字列の四文字を1単位とし，英語文字列の母音字と子音字の並び方からカタカナ1音に対応する文字群へ変換する．評価方法は，変換結果と適応データのカタカナ表記と比較し，正しい変換かどうかを判断する．ここでカタカナ表記のゆれを考慮しており，次の条件がある．それは，(1) 完全一致のもの，(2) 1文字違いのもの，たとえば「ティー」と「テー」,「ショウ」と「ショー」など，(3) 2文字以上違うもの，たとえば“di”に対して「ジ」と「ディ」，“du”に対して「ダ」と「ドゥ」などである．上記の3つの条件を満たすものを正しい変換とし，全体での正しい変換率を求めている．この手作りのカタカナ変換テーブルを用いた方式での正変換率は，人名882個で81.1%であった．

規則そのものを利用することはできないので，評価方法を文献に合わせることにする．まず，こちらで生の情報に混入している誤りを人手で訂正した．さらに，綴りの復元で置換え規則の選択が誤っていた場合に，検索システムは置換え規則を取り替えて実行できることを考え，これらのデータに対して，手で規則を正しく選択し直した．また，表記のゆれの範囲に読みが再現できるものは正解にした．つまり，対応規則の読みの復元率は，カタカナ表記のゆれを考慮していないので，上記の条件(1)から(3)に当てはまるデータを正しく読みが復元できたデータとした．

綴りを復元できたデータ87個に対して，評価を行った結果，(1)は63個，(2)は1個，(3)は6個であった．したがって，本方式での正変換率は80.4%となる．このことより，本方式は人間の知識を利用しないにもかかわらず，人による規則生成と同程度の結果を得ることができたことになる．このことは，生成した規則に人間が考えるであろうような規則が含まれていることを意味する．

6. 考　　察

言語知識の利用と非利用について　本研究では，対応規則の抽出において外国語に依存する知識を使用しないという方針をとったが，言語知識を利用することで抽出の間違いを減らすことができる．一般に，言語知識を利用する方法はいくつかあるが，知識を利用すると，その分析のコストは高い．

ここで，取り扱う言語は多言語にわたるので，言語ごとの知識を獲得しなければならない．それでは，対象の言語を広げるたびに作業が必要となるという問題がある．また，名前のように対象言語を特定すること

すら難しいという状況もある．これらの問題から，できる限り知識非利用の方針を採用した．

一方で，取り扱うデータは日本語に対する外国語というデータで，日本語が中心であった．対象の言語は不特定でも，一方の言語は日本語である．したがって，日本語に関しては言語や発音の知識を分析してアルゴリズムに組み込むとしても，対象言語が増加したり，特定できない状況でも問題はない．

他言語を中心とした場合の閾値の変化 本アルゴリズムに用いた閾値の値は 1/3 である．この値は日本語に依存している．なぜなら，パラメータ設定の理由に述べたように，日本語の性質から得た値だからである．したがって，他言語が中心のデータベースにおいては，パラメータを設定し直す必要がある．

アルゴリズムの性能低下の要因 3.4 節において，本アルゴリズムは，仮想対応規則の 2 段合成データ集合から仮想対応規則を完全に再現できることを示した．しかし，3 段合成データ集合からは仮想対応規則を再現したものの，それ以外の規則も抽出した．これは対応規則が 1 対 1 対応でないことに起因する．仮想対応規則は (n, ン)，(ya, ヤ)，(nya, ニャ) のような規則であり，これは 1 対 1 対応ではない．このような規則の 3 段合成データでは，同じ綴りでも読み方が違うデータができる．たとえば，“nya” は「ニャ」とも読めるし「ンヤ」とも読める．この問題は重要で，一般的なデータにおいても発生する．

EM アルゴリズムとの関係 機械翻訳で外来語のカタカナ翻訳を行う規則の生成[8]という問題もある．規則の多様性を求める本稿の問題とは問題設定が異なるが，類似のデータを扱うという意味で対比する．EM (Expectation Maximization) アルゴリズムによって，最も自然な対応規則の分割を確率の分析で求めることが行われている．頻度は，確率の推定に使われるデータであるので，ここで述べたアルゴリズムも自然な対応規則の分割を確率によって求めるものと解釈できる．また，EM アルゴリズムは音声認識や形態素解析で広く利用されている．

EM アルゴリズムは音素列のような短い列を対象にすることが多い．それを検索に使用すると，検索する範囲が広がりすぎる．ここで提案するアルゴリズムは，取り扱う文字列の長さを長くして EM アルゴリズムを適用する方法に類似していて，その方法でも目的とするような対応規則を生成できる．しかし，解の選択方針において，提案するアルゴリズムは一般的な EM アルゴリズムと異なる．

EM アルゴリズムはもっともらしいものを集中して選択していくアルゴリズムであり，提案する方法は，ある条件を満たすものは選んでいくアルゴリズムである．EM アルゴリズムと比較すると，(1)「抽出した規則の信頼度が保証されていること」，また，(2)「最小の信頼度を同じように補正した場合には，得られる規則の数が多いこと」が違いとなる．

最大の確率を与えるものを選ぶという条件では，ダイナミックプログラミングの方法を使うことで，多分割の問題も効率的に解くことができる．しかしながら，最も自然な分割に限定することという方針は，情報検索で使用するための漏れの少ない規則の抽出という目的にはそぐわない．

EM アルゴリズムによって得られる規則は，少数の規則で自然な読みを与えるという選ばれた規則であり，カタカナから人間が自然に想像するアルファベット列となるような規則である．情報検索にこれを使用すると，人間が推定するのと同じような規則を生成する．これは，人間が行うのと同じ検索を機械に行わせる方法となっている．人間には不得意な検索，つまり，思いがけないが正しい規則によるアルファベット列を検索するという目的には，提案するアルゴリズムの「読みを再現する」という性質が有用である．

対応規則の読みの復元率の値について 読みの復元率の値がデータベース検索の性能に相当すると考えられる．提案するアルゴリズムが示す値は絶対の値として高いのか低いのかは明確ではない．一見では，十分ではない値のようにも解釈できるが，以下のような理由により我々は十分な値と解釈している．

まず，データベースでの検索に使うことが目的であるので，読みの復元率の検査は厳しく行った．しかし，間違いと判定したものを調べてみると，人間ならばこうも読むだろうという結果がいくつかあった．たとえば，人名辞書においてアルファベット表記 “caine” に対する読みは「ケイン」であるが，復元したデータの読みは「カイン」や「キャイン」などであった．

また，対応規則の正当率が 80%であっても，個々の対応規則を合成したものは，$80 \times 80 = 64\%$，$80 \times 80 \times 80 = 51.2\%$にはならない．これは，規則の合成が難しいことを示している．つまり，対応規則を求めることが達成できても，読みの復元は完全にできない．対応規則による読みの付与という方法の根本的な限界が読みの復元率の結果に影響している．

評価で人手の規則と対比したように，提案する方法は人の生成するような規則を含んでいると考えられ，この読みの復元率は自動生成のシステムとして有効性のあるものと考えるが，さらに改良する要求が生じる

値である．

7. おわりに

本稿では，情報検索で使用することを念頭に置いて，外国人名のアルファベット表記と日本語読みが対応したデータの集合から，アルファベット表記と日本語読みの対応規則を自動的に抽出するアルゴリズムを提案した．

本アルゴリズムは，データ集合におけるアルファベット表記の部分文字列とカタカナ表記の部分文字列がともに出現する頻度に注目し，日本語の知識2つと出現頻度を調べる2つのパラメータにより対応規則を抽出する．取り扱うデータは外国語に対する日本語とし，パラメータの値を日本語に依存した値に設定し，その実験値を求めた．また，ローマ字綴りからなる仮想対応規則を作成し，それを合成したデータ集合から仮想対応規則が完全に再現されることを確認した．本方式で得られる対応規則を，規則としての正しさを示す正当率と，規則としての復元率である綴りの復元率と，アプリケーションの性能である読みの復元率とで評価した．実験データは，西洋人名のアルファベット表記と日本語読みが対応したデータ約32000個を用いた．アルゴリズムを適用して抽出される対応規則を評価するためにナイーブな対応規則を用意し，比較実験を行った．抽出した対応規則の正当率は80%，綴りの復元率は84%，読みの復元率は48.8%であった．また，実験データの言語を区別した3種類のデータ，各2400個に対しても同様の評価を行った．この評価の結果，単純な手法により情報検索で効果がある関係を抽出している．さらに，方式の比較として，人手によって規則を作る方法との比較を行った．この結果，本方式は人の知識を利用しない方式であるにもかかわらず，人間の知識を利用した対応規則と同様な関係を抽出している．

今後，現在は単にヒューリスティクスとして記述されているアルゴリズムに確率の見地から定式化を行いたいと考えている．特に，確率モデルによる最尤推定の考え方との対比を明確にして，理論的により性能の高いヒューリスティクスを探したい．

謝辞 本研究のきっかけを与えてくださったNTTヒューマンインタフェース研究所（現北陸先端大）の嵯峨山茂樹氏，ATRの塚田元氏に深く感謝いたします．本研究はNTT基礎研究所の工学研究育成の援助を受け，IPAの独創的先進的情報技術にかかわる研究の一部として，住友電気工業株式会社との共同研究の過程で生まれた成果です．また，適切な査読をしてくださった査読者に敬意を表します．

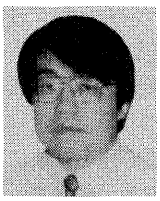
参考文献

1) 国立国語研究所：日本語教育指導参考書16 外来語の形成とその教育，大蔵省印刷局 (1990).
2) 宮内忠信：カタカナ表記からの英単語検索システムの実現，情報処理学会自然言語処理研究会報告，NL-97-17 (1993).
3) 堀内雄一，山崎一生：英単語のアルファベット表記から仮名表記への変換，情報処理学会自然言語処理研究会報告，NL-79-1 (1990).
4) 塚田　元，増田恵子：英単語に対する日本語読み付与方法の検討，第53回情報処理学会全国大会論文集，3-359 (1996).
5) 住吉英樹，相沢輝昭：英語固有名詞の片カナ変換，情報処理学会論文誌，Vol.35, No.1, pp.35–45 (1994).
6) 星野　裕，加藤博子，永田健児：8万人西洋よみ方綴り方辞典，日外アソシエーツ (1994).
7) 新村　出：広辞苑第四版，岩波書店 (1991).
8) Knight, K. and Graehl, J.: Machine Transliteration, *ACL97*, pp.128–135 (1997).

(平成10年3月13日受付)
(平成11年5月 7 日採録)

増田　恵子

1973年生．1996年豊橋技術科学大学工学部情報工学課程卒業．1998年同大学大学院工学研究科修士課程情報工学専攻修了．同年日本電信電話（株）入社，企業システム関連業務に従事．ネットワーク関連システムに興味を持つ．



梅村　恭司（正会員）

1959年生．1983年東京大学工学系研究科情報工学専攻修士課程修了．同年日本電信電話公社電気通信研究所入所．1995年豊橋技術科学大学情報工学系助教授，現在に至る．博士（工学），システムプログラム，記号処理の研究に従事．ACM，ソフトウェア科学会，電子情報通信学会，計量国語学会各会員．