自然言語処理 120－15
(1997. 7. 25)

人名辞書から名前読み付与規則を抽出する試み

増田 恵子† 梅村 恭司‡

豊橋技術科学大学 情報工学系 梅村研究室
〒441 愛知県豊橋市天伯町雲雀ヶ丘 1-1
Tel. 0532-47-0111(内線 5430)
† masuda@avenue.tutics.tut.ac.jp
‡ umemura@tutics.tut.ac.jp

あらまし

人名のアルファベット表記とカナ表記からなる辞書が存在する．アルファベット表記とカナ表記の対応規則は，日本人が外国人名を検索する時に有用である．我々は人名辞書からアルファベット表記とカナ表記の対応規則を抽出することを試みた．本稿では，対応規則を抽出するアルゴリズムを提案し，データとして人名辞書をアルゴリズムに適用し，抽出した規則が辞書のアルファベット表記をどのくらい補えるかの確認を行なった．我々の提案するアルゴリズムは，アルファベット表記に対応するカナ表記を与えずに規則を抽出することができる．

キーワード 形態素，読み，コーパス，統計的言語処理

Extracting Kana-alphabet rules from a non-Japanese name reading table

Keiko Masuda† Kyoji Umemura‡

Umemura Laboratory, Department of Information and Computer Sciences
Toyohashi University of Technology
1-1 Tempaku Toyohashi Aichi 441 Japan
Tel. 0532-47-0111(Ext. 5430)

Abstract

We have a dictionary that converts alphabet of person names into their corresponding Kana. The correspondence rules between alphabet and Kana are valuable when Japanese people search foreign names. We have tried to extract these rules from the dictionary. In this paper, we propose the algorithm that can extract the rules. Then we have applied the algorithm to the actual data. Finally, we have verified the correctness of the rules against original dictionary. It is remarkable that this algorithm is able to extract rules without any given information of Kana-alphabet correspondence.

key words morphology, corpus, statistical processing

1 はじめに

人名辞書と呼ばれる外国人名のアルファベット表記とカナ表記が対応した辞書があり，これは電子化されている．日本人がこの人名辞書を用いるときは，アルファベット表記の読み情報をカナ表記から得たり，カナ表記からアルファベットの綴りを調べたりするだろう．外国人ならばその逆の使い方で，アルファベット表記からカナ表記を得て，カナ表記から本来のアルファベット表記を知るだろう．外国人にとってアルファベット表記をカナ表記にすることは難しいという指摘がある [1]．これは日本人が複数の外国語の音を少数の日本語の音に置き換えるため起こるのであろう．同様に日本人にとってもカナ表記の外国語をアルファベット表記にすることは，アルファベット表記の元の発音情報が落ちているために難しい．このような理由から人名辞書は利用価値がある．

人名辞書が有効であることは述べたが，アルファベット表記とカナ表記との対応をひとつひとつ手で登録するのは大変であるしデータ量も多くなる．そこで英単語のアルファベット表記とその発音記号を用いてカタカナ表記を生成する手法が提案されている [2]．またカタカナ表記には表記のゆれがあるため，そのゆれを統一する手法も提案されている [3][4]．そしてカナ表記のゆれを統一することでデータ量を圧縮できる．

日本人は外国語のカナ表記を知っていてもアルファベット表記を知らないということがあることは先に述べた．このことから，辞書検索においてカタカナ表記を使った英単語の検索が行なわれている [5]．我々は人名を検索するときにはアルファベット表記からの検索だけでは役に立たず，カナ表記での検索も必要だと思っている．従ってアルファベット表記とカナ表記との対応規則が必要である．

これまでの英単語をカナ表記にする手法では，英単語のアルファベット表記と発音記号の情報からカナ表記を得ていた．しかし発音記号の情報を用いずにカナ読みを得る方法も提案されている [6]．我々が取り扱うデータは人名辞書である．日本人が外国人名のアルファベット表記を読むときには表記をそのまま日本語に当てはめて読んでいる場合が多く [1]，カナ表記を使用するのが実情に合致していると考えた．そこで我々は発音記号を用いずにアルファベット表記とカナ表記が対応する規則を抽出することを試みる．

本稿では外国人名のアルファベット表記と日本語の読みであるカナ表記とを持つ人名辞書から，アルファベット表記とカナ表記の対応規則を抽出するアルゴリズムを提案する．本アルゴリズムは人名辞書におけるアルファベット表記とカナ表記の出現頻度をもとに対応規則を抽出する．我々は抽出した対応規則がどれだけ元の人名辞書を復元できるかという実験を行ない，その結果をもとに本アルゴリズムを評価した．

2 問題の定義

アルファベット表記に対応するカナ表記があるとき，対応する2つのペアを対応規則と呼ぶことにする．本稿では人名辞書から対応規則を抽出することを試みる．人名辞書は西洋人名の名字あるいは名前のアルファベット表記とカナ表記が対応したデータである．

カナ表記は日本語の音節構造である．子音をC，半母音をS，母音音素をVとすると，日本語の音節は(C (S)) Vの式で表すことができる．()は省略可能なことを示す．例えば母音(V)「ア・イ・ウ・エ・オ」と直音(CV)「カ・キ・ク・ケ・コ」など，そして拗音(CSV)「キャ・キュ・キョ」などを表すことができる．また特殊音素である促音(つまる音)や引く音素(伸ばす音)は語頭にはこない性質があるので，(C (S)) V ＋ 特殊音素を一つの音節と考えることにする．例えば促音「カッ・キッ・クッ・ケッ・コッ」などや引く音素「カー・キー・クー・ケー・コー」などを表す．ここで撥音(はねる音)「ン」も特殊音素であるが「ン」は一つの音節と考える．このようにカナ表記では日本語の一音節を一つの塊として表現する．

カナ表記は日本語の音節構造に対応したが，アルファベット表記では取り扱う人名が西洋人名のため，英語表記やドイツ語表記といった複数の表記が含まれる．従ってアルファベット表記では特に音節単位に対応しない．このように区切りの違う2つのデータの大きな辞書があるという前提で辞書の単語対単語の情報から音の対応規則の候補を取り出すのが本稿の問題である．

我々は人名辞書のアルファベット表記とカナ表記が対応したデータから尤もらしい対応規則を求めたい．そこで1つの人名データのアルファベット表記を2つにわけたとき，それぞれに対応する尤もらしいカナ表記を求めることで対応規則を得ることにする．人名のアルファベット表記を $A=a_1a_2\cdots a_n$ と表し，a_i $(1\leq i\leq n)$ はアルファベット表記の各文字で n は文字数を示す．同様にカナ表記を $K=k_1k_2\cdots k_m$ と表し，k_j $(1\leq j\leq m)$ はカナ表記の各音節で m は音節数を示す．

始めに人名辞書の各人名データを前半と後半に分けて対応規則の候補を作る．$pair(A,K)$ はアルファベット表記 A とカナ表記 K が対応することを示し，$\{F,R\}$ は F が人名を2つに分けたときの前半の対応規則候補，R が後半の対応規則候補であることを示す．1つの人名データから対応規則を求めるために得られる規則候補を以下のように表す．

前半候補　$F_{ij} = pair(a_1 \cdots a_i,\ k_1 \cdots k_j)$
ただし$a_1 \cdots a_1$はa_1と同じ，kについても同様
後半候補　$R_{ij} = pair(a_{i+1} \cdots a_n,\ k_{j+1} \cdots k_m)$
$1 \leq i \leq n-1,\ 1 \leq j \leq m-1$

例えば人名のアルファベット表記が$a_1a_2a_3$でカナ表記がk_1k_2であるとき，対応規則の候補は

$\{\ F_{11} = pair(a_1,\ k_1),\quad R_{11} = pair(a_2a_3,\ k_2)\ \}$,
$\{\ F_{21} = pair(a_1a_2,\ k_1),\quad R_{21} = pair(a_3,\ k_2)\ \}$

で表される．我々は，アルファベット表記Aとカナ表記Kが与えられたとき，F_{ij}とR_{ij}から尤もらしい対応規則を選択するアルゴリズムを記述する．

3　アルゴリズム

アルファベット表記Aとカナ表記Kが与えられているときその長さをそれぞれn，mとすれば，前半候補F_{ij}と後半候補R_{ij}はともに$(n-1)(m-1)$個できる．F_{ij}とR_{ij}で定まる規則候補について，人名辞書データでの出現度数を計数する．もし前半候補F_{ij} $(1 \leq i \leq n-1,\ 1 \leq j \leq m-1)$のアルファベット表記$a_1 \cdots a_i$がある人名データのアルファベット表記$A$(の一部)と一致したなら，それぞれのアルファベット表記に対応するカナ表記に対して操作をする．すなわち前半候補F_{ij}のカナ表記$k_1 \cdots k_j$と，人名データのアルファベット表記Aに対応したカナ表記Kとの一致を調べる．後半候補R_{ij} $(1 \leq i \leq n-1,\ 1 \leq j \leq m-1)$についても同様の処理をする．上記の操作を全人名データに対して行ない，F_{ij}とR_{ij}の全ての規則候補に対して一致した数をカウントする．$M(R)$は規則候補Rが人名辞書の人名データに一致した数を表すことにする．このとき前半候補F_{ij}の一致数は$M(F_{ij})$，後半候補R_{ij}の一致数は$M(R_{ij})$，$1 \leq i \leq n-1,\ 1 \leq j \leq m-1$で表される．このとき1つの人名に対して作成された前半候補の一致数$M(F_{ij})$の2次元配列M_F，後半候補の一致数$M(R_{ij})$の2次元配列M_Rは以下のようになる．

$$M_F = \begin{bmatrix} M(F_{11}) & \cdots & M(F_{1(m-1)}) \\ \vdots & \ddots & \vdots \\ M(F_{(n-1)1}) & \cdots & M(F_{(n-1)(m-1)}) \end{bmatrix}$$

$$M_R = \begin{bmatrix} M(R_{11}) & \cdots & M(R_{1(m-1)}) \\ \vdots & \ddots & \vdots \\ M(R_{(n-1)1}) & \cdots & M(R_{(n-1)(m-1)}) \end{bmatrix}$$

アルファベット表記$A = a_1a_2a_3a_4$，カナ表記$K = k_1k_2k_3$のときのM_Fを図1で表す．図1のM_Fの1行目はカナ表記k_1に対応するアルファベット表記がそれぞれa_1の時の一致数，a_1a_2の時の一致数，$a_1a_2a_3$の時の一致数を示す．

	k_1	k_1k_2
a_1	$M(F_{11})$	$M(F_{12})$
a_1a_2	$M(F_{21})$	$M(F_{22})$
$a_1a_2a_3$	$M(F_{31})$	$M(F_{32})$

図 1: $A = a_1a_2a_3a_4, K = k_1k_2k_3$での$M_F$

このようなM_F, M_Rを人名辞書の全データに対して求め，M_F, M_Rに見られる一致数の頻度をもとに候補の中から対応規則を選ぶ．定義より下の性質があり，本アルゴリズムはこれを使用している．

1. (a) F_{ij}は対応規則の前方候補であるので，人名辞書の全データと比較したとき$M(F_{ij})$はi,jについて単調に減少する．
   (b) R_{ij}は対応規則の後方候補であるので，人名辞書の全データと比較したとき$M(R_{ij})$はi,jについて単調に増加する．
2. $M(F_{ij})$と$M(R_{ij})$は，添字iとjの数が近いほど大きくなる．

この2つの性質をもとにF_{ij}とR_{ij}の対応規則候補の中から尤もらしい対応規則を選択するアルゴリズムを示す．通常人名データはアルファベット表記の長さよりカナ表記の長さの方が短い．そこで本アルゴリズムでは各対応規則候補のカナ表記の部分を固定して，それに対応するアルファベット表記の部分はどこまでかということを検証していく．例えば図1のM_Fをアルゴリズムに適応するならば，k_1に対応するアルファベット表記はa_1か，それともa_1a_2か，それとも$a_1a_2a_3$かということを検証していく．

アルゴリズム

1. (a) 前後の比$R_M(F_{ij}) = M(F_{(i+1)j}) \,/\, M(F_{ij})$, $1 \leq i < n-1,\ 1 \leq j \leq m-1$とする．$M_F$の行を順に見るとき，$M(F_{ij})$が定数$C$を持ち$R_M(F_{ij})$が閾値$TH$となったときの$Fij$はその行の前方対応規則である．
   (b) 前後の比$R_M(R_{ij}) = M(R_{(i-1)j}) \,/\, M(R_{ij})$, $n-1 \geq i > 1,\ m-1 \geq j \geq 1$とする．$M_R$の行を順に見るとき，$M(R_{ij})$が定数$C$を持ち$R_M(R_{ij})$が閾値$TH$となったときの$Rij$はその行の後方対応規則である．

2. $i \simeq 2/n$, $j \simeq 2/m$ に近い添字を持つ F_{ij}, R_{ij} を有効とする.

ここで定数(Const)C と閾値(THreshold)TH の由来とアルゴリズムでの値を述べる. まず C だが, 我々が用いた人名データには誤り(サンプル調査で1%程度)があるので, 統計的に不安定となるデータを除かねばならない. そこで今回の実験では $C = 10$ を選び, これ以上の一致数を持つものを選択することにした.

次に TH の由来である. 次のような人名データのモデルを考える. それは人名データのアルファベット表記がアルファベット文字の前後関係によらないモデルである. 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. このモデルからある M_F と M_R が得られたとする. この M_F と M_R をアルゴリズムに適応すると, 規則候補の一致数は行毎で添字が大きくあるいは小さくなるに従ってなだらかに減少するはずである. しかし実際の人名データでは違う. 実際の人名データから得た M_F と M_R をアルゴリズムに適応する. このとき規則候補の一致数は行毎で添字が大きくあるいは小さくなるに従って減少していく. しかし注目しているカナ表記の部分に対してアルファベット表記の部分が対応する可能性がなくなったとき, その規則候補の一致数は上記に仮定したモデルでの値より急激に減少する. つまり規則候補の一致数は人名データを2つに分けたときの本来の切れ目でない所にくると急激に減少する. 後にその実例を示す. この考え方が本アルゴリズムの着眼点である. この切れ目はアルファベットの数などにも依存するが, 今回の実験では $TH = 1/3$ とした.

次章で本アルゴリズムを人名辞書に適応した結果を示す.

4 アルゴリズムの適応結果

人名辞書のデータは西洋人名のアルファベット表記とカナ表記が対応した約71000人分のデータである. アルファベット表記にはアルファベット26文字以外に変音文字(ü, é等)60文字を含んでいる. またアルファベット表記とカナ表記とが対応していない人名データ(ノイズ)を1%程度含む. 前章で説明したアルゴリズムをこの人名辞書のデータに適応した. この実例を1つ示す.

人名辞書のデータがアルファベット表記 $A =$ abramov, カナ表記 $K =$ アブラモフ で, それぞれの長さ $n = 7$, $m = 5$ のときの前方規則候補の2次元配列 M_F と後方規則候補の2次元配列 M_R を図2に示す. 数字の上あるいは下にあるラインは $TH = 1/3$ になる場所を示す. この M_F と M_R にアルゴリズムを適応する. M_F の行を左から順に見ていくと, F_{11}, F_{22}, F_{43} の3つ

$$M_F = \begin{bmatrix} \underline{3657} & 124 & 28 & 5 \\ 220 & \underline{105} & 25 & 4 \\ 30 & 29 & 23 & 4 \\ 23 & 23 & \underline{23} & 4 \\ 7 & 7 & 7 & 4 \\ 5 & 5 & 5 & 4 \end{bmatrix}$$

$$M_R = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 2 & 3 & 5 & 5 \\ 2 & 3 & 8 & 8 \\ 2 & 3 & \overline{45} & 45 \\ 2 & 3 & 45 & \overline{1085} \\ 2 & 3 & 45 & 1666 \end{bmatrix}$$

図 2: $A =$ abramov, $K =$ アブラモフ の M_F と M_R

が前方対応規則の候補であることがわかり, アルゴリズムの2において F_{22}, F_{43} の規則が残る. 同様に M_R の行を右から順に見ていくと, R_{54}, R_{43} の2つが後方対応規則の候補であることがわかり, アルゴリズムの2において R_{43} の規則が残る. アルゴリズムを適応した結果, 人名データがだいたい半分に別れるような次の前方対応規則と後方対応規則が残る.

$$F_{22} = pair(\text{ab}, \text{アブ})$$
$$F_{43} = pair(\text{abra}, \text{アブラ})$$
$$R_{43} = pair(\text{mov}, \text{モフ})$$

抽出した前方あるいは後方対応規則は人名を2つに分けたものの片方なので, 残りの片方も対応規則として追加する. 従って対応規則は以下のようになる.

{ **pair(ab, アブ)**, $pair$(ramov, ラモフ) },
{ **pair(abra, アブラ)**, $pair$(mov, モフ) },
{ $pair$(abra, アブラ), **pair(mov, モフ)** }

同じ対応規則が複数できることになるが, 今回の実験では別の対応規則としてカウントしている.

このようにして人名辞書のデータから対応規則を抽出した. この結果得られた対応規則の数を表1に示す.

対応規則	F_{ij}	R_{ij}
抽出数	25152	28703
全抽出数	53855	

表 1: 抽出した対応規則数

5 対応規則の評価

我々はアルゴリズムを適応して抽出した対応規則を評価するために，基準となるナイーブな対応規則を用意した．それは人名辞書のアルファベット表記とカナ表記をそれぞれ半分ずつに分けて対応を取ったものである．我々のアルゴリズムで抽出した対応規則を対応規則α，ナイーブな対応規則を対応規則βと呼ぶことにする．

対応規則には1つのアルファベット表記に対応するカナ表記が幾つかある場合がある．そこで対応規則をアルファベット表記でソートし，1つのアルファベット表記に対して複数のカナ表記が当てはまるようにした．(従って対応規則αの数は前章で示した数より少なくなる．)

評価実験1

対応規則αとβから100個を無作為抽出し対応規則が正しいかどうかを調べる．

1つのアルファベット表記に対応するカナ表記が幾つかある場合にはその中の一つがあっていればその読み対応は正しいと見なした．この結果正しかった規則数を表2に示す．

対応規則	α	β
規則数	76	54

表 2: 対応規則の調査結果

評価実験2

対応規則αとβが対応規則の抽出に使用しなかった人名データにおいて，どれくらい人名の読みを復元できるか調べる．つまり読みの対応がどのくらい取れているかを調べる．

対応規則の抽出に使用しなかった人名データ100個に対して対応規則αとβを当てはめる．人名データの復元には，具体的には人名データと対応規則との前方からのマッチングを行なった．つまり，人名データのアルファベット表記と対応規則のアルファベット表記との前方からの一致を見ていき，もし一致したときには人名データのアルファベット表記を対応規則のカナ表記に置き換えるという作業を行なった．この実験結果を表3に示す．表の完全置換数とは，人名データのアルファベット表記を完全に対応規則のカナ表記で置き換えられた人名データ数である．

評価実験3

対応規則αとβを当てはめた人名データの読みが正しいかどうかを調べる．

対応規則	α	β
対応規則数	26375	30373
完全置換数	67	100

表 3: 人名データの復元実験結果

評価実験2において対応規則αとβで人名データを完全に置き換えることのできたそれぞれのデータから(対応規則αが67個，βが100個)，人名の読みが正しいものを調べた．この結果読みが正しかった人名データの数を表4に示す．

対応規則	α	β
人名データ数	45	51
(割合)	(67%)	(51%)

表 4: 人名データの読み調査結果

実験の結果，対応規則αを当てはめた人名データにおいて読みが正しかったものには，人名データを3つあるいは4つの対応規則で置き換えているものがあった．しかし対応規則βを当てはめた人名データにおいて，人名データを3つあるいは4つの対応規則で置き換えているものに読みが正しいものはなかった．

6 考察

前章の実験結果から抽出した対応規則を考察する．

評価実験1の結果より，対応規則αの約76%が正しいことがわかった．我々の提案するアルゴリズムは有効であると言えよう．

評価実験2では対応規則αより対応規則βの方が一見すると良い結果である．対応規則αにおいて人名データが置き換えられなかったものは長い人名である．事実として人名データのアルファベットの長さnが長くなるほど$n/2$付近では他の人名データとの重なりが少なくなる．今回の実験ではこのような人名データで対応規則を抽出するときに$n/2$付近の対応規則候補の一致数がアルゴリズムの条件Cを満たさない場合が増加した．これを改善するにはもっと多くの人名データが必要である．

評価実験3の結果より，対応規則αは67%，対応規則βは51%の人名データにおいて正しく読み付与することができた．先にも述べたように，対応規則αは人名辞書から抽出することのできる対応規則が少ない．しかしそれでも対応規則αが人名データを3つあるいは4つの対応規則で読み付与することができたことに

注目すべきだろう.

7 今後の課題

以下の課題があり検討していく予定である.

- 現在の対応規則は1つのアルファベット表記に対して複数のカナ表記が存在する. 実際にはどのカナ表記を選ぶのが適切かということを検討する必要がある.
- 今回のアルゴリズムでは, アルファベット表記とカナ表記をそれぞれ2つに分割して読み対応規則を抽出した. さらに細かい対応規則を得るためには, 抽出した対応規則を分割する必要があるだろう. しかし対応規則の中にはさらに分割したほうがよい規則とこれ以上分割しなくてもよい規則があると思われる. これをどう見極めるかが問題である.
- 我々の提案するアルゴリズムは, 考察でも述べたが人名のデータ量が少ないと抽出される対応規則が少なくなる. 従ってもっと多くの人名データを用いて実験する必要がある.
- はじめに述べたようにアルファベット表記での検索だけでなくカナ表記での検索は有効である. そこで抽出した対応規則を用いて人名検索を行ない, 対応規則の効果を確認する必要がある.
- 我々の提案するアルゴリズムは2つの対応する関係からその部分関係を抽出する. このアルゴリズムが人名辞書以外に適応できるかを検証することも将来の課題である.

8 関連研究

我々の提案するアルゴリズムは, 人名辞書中のアルファベット表記とカナ表記の出現頻度をもとに対応規則を抽出する. もととなるデータやレベルは異なるが, 対訳テキストコーパスを用いて同じような取組みがなされているので紹介する.

1つめは対訳テキストコーパスの二言語間の文字の長さを元に比較し, 対訳表現を並べるものである [7]. 二言語間の単語の対応情報をまったくもたずに行なっている所が我々の手法と同じである. しかし対応づけているレベルが異なる.

2つめは対訳テキストコーパス中の二言語間の単語の共起頻度を用いて対訳表現を自動抽出するものである [8]. 2つの対応をとるという考えにおいて, 共起に注目しているのは共通のアプローチである. しかし1つめと同様に対応づけているレベルが異なる.

9 おわりに

我々は人名辞書からアルファベット表記とカナ表記との対応規則を抽出するアルゴリズムを提案した. 本アルゴリズムで抽出した対応規則を用いて人名データがどのくらい復元できるかの実験を行なった. この結果, 人名データを76%復元できた. また復元した人名データの読み付与は67%があっていた. 我々の提案するアルゴリズムは, アルファベット表記に対応するカナ表記を与えずに規則を抽出することができる.

謝辞

本研究の種を与えて下さったNTTヒューマンインタフェース研究所の嵯峨山茂樹氏, ATRの塚田元氏に感謝致します. この研究はNTTより工学助成の寄付金による補助を得ています.

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] 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 79-1, 1990.

[3] 獅々堀正幹, 青江順一：カタカナ異表記の生成および統一方法, 情報処理学会 自然言語処理研究会報告 94-5, 1993.

[4] 久保田淳市, 庄田幸恵, 河合眞宏, 玉川博文, 杉村領一：カタカナ表記の統一方式, 情報処理学会 自然言語処理研究会報告 97-16, 1993.

[5] 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 97-17, 1993.

[6] 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, 情報処理学会第53回全国大会 3-359, 1996.

[7] William A.Gale, Kenneth W.Church, A Program for Aligning Sentences in Bilingual Corpora, Association for Computational Linguistics, 19(1), pp.75-102, 1993.

[8] 北村美穂子, 松本裕治：対訳コーパス中の共起頻度に基づく対訳表現の自動抽出, 情報処理学会 自然言語処理研究会報告 114-11, 1996.