

# 携帯電話における日本語入力 ——子音だけで日本語が入力できるか

田中 久美子<sup>†</sup> 犬塚 祐介<sup>††</sup> 武市 正人<sup>†††</sup>

本稿では、携帯電話のための子音漢字変換による日本語入力方式を議論する。子音漢字変換では、ユーザが子音列を入力し、これを子音漢字変換ソフトウェアが漢字混じりの日本語に直接変換する。従来のかなを入力する方式では1かなあたり複数回キー入力を行う必要があるのに対し、子音入力を用いると、1かなあたり1回だけの入力済む。一方で、子音漢字変換を行うと、かな漢字変換の7倍にもものぼる候補を扱わざるをえない。そこで、本研究においては、隠れマルコフモデルを用いた言語モデルに PPM (Prediction by Partial Match) を融合させた動的な言語モデルを用いてユーザの語彙選択の傾向を学習させることにより、入力効率を追求した。その結果、現行のかな入力方式に対して打鍵数を半減させることができた。ユーザ実験を行った結果でも、現行の方式よりも短時間で入力を行うことができることが分かった。

## Japanese Text Entry with Digits ——Can Japanese be Estimated Only From Consonants?

KUMIKO TANAKA-ISHII,<sup>†</sup> YUSUKE INUTSUKA<sup>††</sup>  
and MASATO TAKEICHI<sup>†††</sup>

We discuss a Japanese text entry method for mobile phones. Different from the current methods based on kana kanji conversion, our system asks user only to enter the sequence of consonants that requires single stroke per kana character. Our consonant-kanji conversion system then converts the sequence directly into the final kanji form. Although such consonant-kanji system should handle seven times more candidates than kana-kanji conversion system, we found that user may enter the same text with half number of keystrokes within shorter time. The key to realize such consonant kanji conversion system lies in our dynamic language model based on Hidden Markov Model and Prediction by Partial Match (PPM) method.

### 1. はじめに

近年では携帯電話が普及し、さらに i-mode や電子メールといったサービスの充実により、携帯電話上でもテキストを入力する機会が増えている。携帯電話上でのテキスト入力では、別売のキーボードを用いる方法もあるが、一般には携帯電話にもともと付いている約 10 のキーだけを使って入力されている。

現行の携帯電話上の日本語入力は、かなを入力してこれを漢字に変換することによる。かなは子音と母音の組で決定されるため、子音と母音を数字に変換し、

これを 10 のキーで指定することにより入力される。主な方式としてかな方式、ポケベル方式といったものがあり、現在は前者が主流である。これらの方式の最大の問題は打鍵数が多くなってしまふ点である。その結果、ユーザに負担がかかり、入力に時間がかかるわりには入力間違いを避けられないうえ、腱鞘炎などの問題が世界的にも問題視されているのが現状である<sup>4)</sup>。

この問題を解決するために、本稿では、キーを押す回数を減らした日本語入力方式を議論する。具体的には、かなを入力する代わりに子音のみを入力し、目的とする漢字へ変換する子音漢字変換方式である。たとえば「日本語」の入力は、従来のローマ字入力であれば「nihongo」と入れるところを「nhng」を入力することになる。数字では、n が 5 (な行), h が 6, n が 0 (わ行), g が 2\* に対応し「5602\*」となる。

このような子音による入力は、現在のパソコン上でのかな漢字変換が確立する前にも提案されたことがあり、本研究が最初ではない<sup>13)</sup>。しかし、その後、パソ

<sup>†</sup> 東京大学大学院情報学環  
Interfaculty Initiative in Information Science, The University of Tokyo

<sup>††</sup> 東京大学大学院工学系研究科  
School of Engineering, The University of Tokyo

<sup>†††</sup> 東京大学大学院情報理工学系研究科  
Graduate School of Information Science and Technology, The University of Tokyo

コンでの日本語入力にローマ字入力へと収束するにしたがって、子音だけをういた入力方式の研究はすたれてしまった。ところが、昨今の携帯電話の普及により、この問題は再び脚光をあびるにいたっている。たとえば、Tegic 9<sup>1)</sup> や ZI<sup>9)</sup> は英語に対して同様のシステムを提案しているほか、日本語入力に関しても類似の入力方法に着手している。また、NTTの案内ジョーズ<sup>6)</sup> では、固有名詞の入力に対してこの入力方式を採用し、電話番号案内を実用化している。これらの先行研究に対し、本稿の入力方式の独自性は2つある。まず、案内ジョーズとは異なり、日本語の通常の文章入力に対して子音入力を適用した点である。第2に、Tegic 9 や ZI は子音入力に対してかな列をユーザが一度決定し、さらにかな漢字変換を行う2段階変換方式である。これに対し、我々のものは子音列から一気に漢字列へ変換を行う1段階変換方式、すなわち子音漢字変換を提唱し、実現に必要な言語モデルを模索した点である。本稿では子音漢字変換に関連する入力方式を網羅的にとりあげて、打鍵数の測定とユーザ実験により、その効率を比較検討した。

子音漢字変換では、入力時に打鍵数を減らすことができる一方で、従来のかな漢字変換と比べ、子音列に対する候補数が増大する問題が生じる。そこで、本稿では、子音漢字変換を概説した後に、この候補数に関する考察を述べる。続いて子音漢字変換を現実的な効率で実現するための、言語モデルおよび学習モデルについて説明する。実際に実装した TouchMeKey システムの概要を示し、最後に効率について従来入力方式と比較検討する。

## 2. 子音漢字変換とその問題点

### 2.1 従来方式と子音入力方式

現在携帯電話で最も普及している日本語入力方式には2つあり、かな方式、ポケベル方式といわれる。これらはいずれも「かな」を携帯電話で入力する。かなを指定するには、子音と母音の両方を携帯電話のキー、すなわち数字により指定する必要がある。子音、母音への数字の割当ては機種ごとに多少の異なりはあるが、おおむね表1のようになっている。たとえば、「ほ」は子音は6、母音は5となる。

かな方式とポケベル方式は母音の入力方法が異なる。「かな方式」では、子音キーを押す回数により母音の番号を入力する。たとえば、「こ」は子音2番の母音5番であるので、2を5回押すこととなる。「はこ」を入

表1 子音と母音の数字割当て

Table 1 Digits corresponding to a consonant or a vowel.

子音				
0	1	2	3	4
わ (w)	あ (a)	か (k)	さ (s)	た (t)
5	6	7	8	9
な (n)	は (h)	ま (m)	や (y)	ら (r)
母音				
1	2	3	4	5
あ	い	う	え	お

力するにはユーザは「622222」と押す。この方式は、ユーザの指の移動は1かなに対して1回であるものの、次のような問題がある。まず、キーを押す回数が非常に多くなり、さらに同じキーを多く押すので、間違いやすい。特に5回押す必要があるときには、間違っただけで6回押してしまうことがある。さらに同じかな文字を連続して入力する場合に問題がある。たとえば、「ひひ」と入力する場合には、6を2+2=4回押すこととなるが、これは同時に「へ」に対応してしまう。このために「ひひ」を入力する際には、最初の「ひ」を入れた後に、別のキーを押して、文字をいったん確定せざるをえない。したがって、連続入力のときだけ特別な入力方式が必要となってしまう。

また「ポケベル方式」(以下PB方式と記述)では、母音の番号のキーをユーザは直接入力する。たとえば、「こ」であれば、子音2の後に母音5を入力する。「はこ」は「6125」と入力される。この場合には、キーを押す回数は、キーボードを用いたローマ字入力方式とほぼ同等となる。しかし、指の移動は1かなに対して2回であるため、携帯電話の小さな10key上ではやはり誤入力を引き起こす。したがって、この方式を用いるユーザは限られている。このため、現在の携帯電話での入力方式は「かな方式」が大半を占める。

これら2つの入力方式を比較検討すると、携帯電話での入力においては、

- 指の移動は1かな1回が望ましい、
  - キーを押す回数も少ないことが望ましい、
- ことが分かる。

そこで、本稿では1かな1押し方式、すなわち子音だけを入力することで文書を入力する方式を取り上げる。たとえば、「はこ」であれば「62」とだけユーザは入力を行うことになる。表2にかな方式、ポケベル方式、本方式で「日本語」を入力するときの例を示す。

ここでの入力も F503i での入力を参考にした。「ご」は「こ」を入力した後、\* を押すことにより得られ、また「ん」に母音はないが、わ行の3番目の母音扱いとなっている。

これは F503i の割当てである。

表 2 各方式で「にほんご」を入力するときのキー操作  
Table 2 Digit sequences to enter “nihongo”.

	に	ほ	ん	ご
かな方式	55	666666	000	22222*
PB 方式	52	65	03	25*
本方式	5	6	0	2*

この方式は上の 2 つの要請を同時に満たしており、ユーザへの負荷は小さくなる。一方で、入力後漢字変換を行う際に、対象となる候補が多くなる。たとえば、「にほんご」を表す「5602\*」は「にほんが」にも対応する。つまり、本方式により日本語を入力する際の問題点は、通常のかなを入力する方式よりも単語候補が増えるため、単語候補をどのように扱うか、という点に集約される。

## 2.2 候補数の見積もり

1 かな 1 押し方式を用いた入力には、1 段階方式と、2 段階方式の 2 つが考えられる。前者は、入力された数字列に対して、漢字変換した結果を直接ユーザに提示する方法である。一方の 2 段階方式は、ユーザが数字列を入力した後、ユーザにかな列の候補を提示して選んでもらい、その後通常のかな漢字変換を用いる。前者の場合は、候補数が多くなるという問題があるのに対し、後者は処理が 2 段階になるという問題があげられる。T9<sup>1)</sup> や ZI<sup>9)</sup> は後者の方式を用いている。しかし、候補の選択には一般に時間がかかるため、できれば 1 段階方式が望ましいであろう。そこで、1 段階方式の場合候補数についてまずは検証する。

たとえば、毎日新聞(94 年)の社会面(総単語数 130 万語、約 28 Mbytes)には異なり語数で約 9 万の単語がある。これら进行かなで表現した場合の平均長を計測したところ、4.81 かな文字であった。従来方式の入力の場合は、1 文字あたり約 50 のかな表現があるので、 $50^{4.81} \approx$  約 1 億 4,290 万語がもともと表現可能である。ところが、単語数は約 9 万であるので、1 億単語中 9 万単語分だけしか使っていないということになる。すなわち、かなによる入力は、もともと余裕を持たせた入力方式となっている。社会面程度の単語数であれば、同音異義語がないと仮定した場合には、1 つの入力に対して、候補数は 1 に絞られることとなる。一方で、本方式では、1 文字あたり 10 の数字表現しかない。単語の平均長はかなの場合と同様で 4.81 数字となるので、表現可能な単語数は、 $10^{4.81} \approx$  約 63,100 語である。とすると、9 万語を表現するには、すべての数字列を使用したとしても、1 つの数字列で

1 つ以上の単語を表現しなければいけないことになる。単語  $W$  の出現確率を  $P(W)$  と記述し、単語  $W$  と子音列  $C$  を共有する単語の総数を  $N_C(W)$  とすると、候補数の平均は、

$$\sum_W N_C(W)P(W) \quad (1)$$

により見積もることができる。毎日新聞の社会面 1 年分を用いてこの値を計算してみると、子音入力の場合は結果は 31.7 語である。一方で  $C$  をかな列とした場合には、値は 4.5 語にすぎない。つまり、子音漢字変換では、かな漢字変換よりも入力における打鍵数が少ない分、候補がじつに 7 倍に増えていることになってしまう。となると、候補をよほど工夫して扱わない限り、実用には耐えない入力システムとなってしまうおそれがある。

確かに本方式では平均候補数は大きい。しかし、平均候補が 31.7 単語であったとしても、それらを適切に並べてユーザに提示することができれば、本方式の実用性を高めることが可能である。たとえば、32 の候補のうち、ユーザが用いるのはほとんど上位の 2 単語のみで、他の 30 の候補はまれにしか出現しない語であれば、この 2 単語を上位に提示すればよい。

さらに、どの単語を頻繁に使うかは、ユーザに依存する。そもそも携帯電話とはユーザにパーソナルな機器である。そこで、ユーザの語彙を学習することにより、さらに実用性を高めることができるであろう。

次章では候補を適切に順序付けるための言語モデルを述べ、その後候補の提示順序をユーザに個別化する方法を述べる。

## 3. 動的な言語モデル

### 3.1 言語モデル

ユーザが入力した数字列を  $C$ 、単語列を  $W$  とすると、最適な単語列を、

$$\hat{W} = \arg \max_W P(W|C) \quad (2)$$

と定義する。右辺はこのままでは扱いにくいので変形する。Bayes の定理を用いると、

$$\hat{W} = \arg \max_W P(W|C) \quad (3)$$

$$= \arg \max_W \frac{P(C|W)P(W)}{P(C)} \quad (4)$$

$$= \arg \max_W P(C|W)P(W) \quad (5)$$

となる。まず、 $P(C|W)$  は  $C$  が与えられているときには、 $C$  の候補となりうる単語  $W$  については値は 1.0 であるのに対し、候補となりえない単語は値は 0.0 で

これは茶筌<sup>12)</sup>で形態素解析した際の単純な異なり語数である。

ある．したがって， $C$  の候補となりうる単語  $W$  だけを考えればよいというあたりまえのことがこの項に現れている．さらにこの式は，これら候補のうち，最適なものは， $P(W)$  が最大となるものであることを示している．そこで，以下では  $P(W)$  の求め方を考える．

本稿では携帯電話を対象機器としている．したがって， $P(W)$  の結果として大きな辞書を必要とするようなものは向かない．たとえば， $P(W)$  を単語の bigram で近似するのでは，その bigram のデータ自体が大きな容量となってしまうことが予想される．

そこで，本稿では品詞列を用いた隠れマルコフモデルを用いて  $P(W)$  を見積もる．これは， $w_i, t_i$  を  $i$  番目の単語およびその品詞として，

$$P(W) \simeq \sum_T \prod_{i=1}^n P(w_i|t_i)P(t_i|t_{i-1}) \quad (6)$$

と近似され，かな漢字変換でも用いられるモデルである<sup>10)</sup>．本稿ではこれを子音漢字変換に応用する．

### 3.2 学習モデル

前節までの  $P(W)$  の値は，毎日新聞など大規模なコーパスから計測するものである．本稿ではさらにこれをユーザコーパスを用いてユーザに特化する．ユーザのコーパスは，一般に量が小さい．小さいコーパスであろうとも，確率の値をコーパスに特化させる方法として，情報圧縮の分野では PPM (Prediction by Partial Match) が提案されている．そこで，本稿では，上記の隠れマルコフモデルと PPM を融合させた動的な言語モデルを提案する．

PPM はもともと算術符号化の圧縮率の向上のために文脈を考慮して確率の補正を行う研究に背景がある．PPM は本入力システム以前にもユーザインタフェース分野のテキスト入力の研究において用いられ<sup>8)</sup>，ユーザの文脈を反映させる目的に効果があることが示されている．以上の点から，PPM による補正を採用した．

まず，PPM を概説する．Bell ら<sup>3)</sup> によると，ある時系列  $S = x_0 \dots x_{i-1}$  が与えられたとき，次の要素  $x_i$  の確率を次式で推定する．

$$P(x_i) = \sum_{k=-1}^{kmax} u_k P_k(x_i) \quad (7)$$

ここで， $k$  とは  $x_i$  の前  $k$  単語のことである． $P_k(x_i)$  は前  $k$  単語を考慮した確率で， $C_k$  をその文脈の現れ

た回数， $c_k(x_i)$  をその文脈で  $x_i$  が現れた回数として，

$$P_k(x_i) = \frac{c_k(x_i)}{C_k} \quad (8)$$

と定義される．特に  $k = -1$  のときは，文脈から情報がまったく得られない場合に基本として参照される初期確率を表現している．

これを我々は次の 2 点において前節の隠れマルコフモデルと融合させる．まず第 1 に，条件付き確率の定義から，式 (6) を変形し，

$$P(w_i|t_i) = \frac{P(w_i, t_i)}{P(t_i)} \quad (9)$$

とする．そして，この  $P(w_i, t_i)$  を PPM により特化する．つまり， $x_i = (w_i, t_i)$  と見なし， $k \geq 0$  の際の  $P_k(w_i, t_i)$  を得る．式 (9) と式 (6) の残りの部分  $\frac{P(t_i|t_{i-1})}{P(t_i)}$  には，大規模なコーパスから得た値をそのまま用いる．むしろ  $\frac{P(t_i|t_{i-1})}{P(t_i)}$  についても，PPM でユーザコーパスから推定することが考えられるが，品詞列は言語モデルの骨格を形成するため，単語列と比較するとユーザに固有な列にはなりにくいと考えられるので，このようにした．

第 2 に，式 (7) における  $P_{-1}(w_i, t_i)$  は大規模なコーパスから得た確率の値に初期化する．これにより，大規模コーパスの確率，つまり単語の一般的な使われ方が候補順位に反映される．

最後に，式 (7) における  $u_k$  について述べる．これは，直前の  $k$  単語に関する確率を最終的な確率値に足し込む重みである．重みの決め方にはさまざまな研究成果がある<sup>7)</sup>．本稿では，PPMA<sup>3)</sup> を用いるものとした．なぜなら，PPMA や PPMC といった学習効果の差も調査はしたが，有意な差は得られなかったからである<sup>2)</sup>．また， $k$  の上限値  $kmax$  は同研究の成果<sup>7)</sup> をそのまま採用し 4 (5-gram) とした．

## 4. 実 装

### 4.1 他のさまざまな入力方式

本稿では従来のかなを入力する方式に対して，子音を入力する方法について論じてきた．これ以外にも，入力方法にはさまざまな選択肢がある．第 1 は，入力する言語単位を単語にするのか，あるいは文節にするのかという点である．第 2 に補完<sup>11)</sup>，すなわち，単語の途中までの入力で，その単語を終わりまで推測するかどうかである．たとえば，子音漢字変換の場合「56」と入力しただけで「日本語」を推定することが補完に相当する．本稿の主旨はこれらの選択肢と相いれないものではない．つまり，かなで入力するのか子音を用

通常自然言語処理でいわれる  $n$ -gram は，PPM では， $k = (n - 1)$  となる．たとえば， $k = 1$  のときは，自然言語処理における bigram を， $k = 2$  のときは，自然言語処理における trigram を意味する．

いるのか、言語の単位として単語か文節か、補完を用いるのか用いないのか、で  $2 \times 2 \times 2 = 8$  通りの入力方法が考えられる。たとえば、かな入力 + 単語単位 + 補完なし、という候補の数が少ない方法も考えられるし、子音入力 + 連文節単位 + 補完ありという候補の数がいかにも多い方法も考えられるであろう。

本稿では、子音入力 + 文節単位 + 補完なし（以下では CP とは記す）、および、子音入力 + 単語単位 + 補完あり（以下では CWC）、の 2 つの入力方法を実装して検討することにした。というのも、我々の主題は子音漢字変換であるため、候補数がすでに多い。このため、連文節と補完をいきなり組み合わせた負荷の高いシステム以前にこの 2 つの実用性を議論すべきであるからである。

ただし、後の実験の際には既存のシステムとの比較を行うため、子音入力 + 単語単位 + 補完なし（CW）という最も単純な方式も考える。また、CP や CWC の比較対象として、かな入力 + 文節単位 + 補完なし（KP）、かな入力 + 単語単位 + 補完あり（KWC）も用意する。

#### 4.2 基本辞書とユーザコーパス

まず、大規模なコーパスとして毎日新聞の社会面を用いて、基本辞書を構築した。これは、3.2 節で説明した  $k = -1$  の場合の初期確率として用いられる。社会面を用いた理由は、その話題が豊富であり、他の新聞記事に比べて単語の専門性や偏りが少ないからである。毎日新聞は形態素解析<sup>12)</sup>、そこに現れたすべての異なり単語から単語辞書を作成した。単語辞書の 1 項目には、

単語、よみ、子音相当の数字列、品詞、頻度が記述されている。社会面 1 年分の異なり単語は前述のように 9 万単語である。

$k = 0$  以上の統計を得るために用いるユーザコーパスは、本稿では 3 種を用意した。毎日新聞の経済面、第 3 著者の電子メールについては 15 万単語、プログラミング言語の教科書については 5 万単語分である。各ユーザコーパスは局所的な偏りをなくすために、まず文章ごとに切り分け、これをランダムに並べ替えた。実際の入力においては文脈は偏っているものであるが、実験上はあるユーザコーパス量の際の平均打鍵数、つまり平均的な効率を計測するためにこのような処理を行った。そのうえで形態素解析を行って、単語、その品詞、対応する数字列の列に変換した。以上の前処理を行ったうえで、学習コーパスとテストコーパスに切り分けた。

学習コーパスは、さまざまな大きさのものを用意し、

大容量のものは小容量のコーパスを含むように作成した。学習コーパスはシステム内部では SuffixArray<sup>5)</sup> として実装した。一方のテストコーパスはこの後の評価実験で用いる自動入力のための文書で、未知語を除いて 1500 単語程度のもものを学習コーパスとは異なる部分から切り出した。

#### 4.3 TouchMeKey システムの概要

TouchMeKey システムは、基本辞書と学習コーパスの 2 つのデータを用いてユーザの入力する子音列を漢字に変換するソフトウェアである。

まず、補完を用いた単語変換の場合には、現在の入力で始まる単語候補を辞書から求め、式 (6) の値の高い順に表示する。

文節変換の場合には、その入力を適当に切り分け、辞書を用いて単語に変換し、候補を生成する。候補は、式 (6) が最大となる候補を永田<sup>10)</sup> の方式を用いて探索する。第 2 候補以下は、最初の単語を同じ長さの別の単語に置き換え、確率が高い順に提示する。これは、候補を単純に確率の順序で提示すると、ユーザは単語の切れ目と単語の選択とを一度に考慮して選択しなければならず、使い勝手がよくない、というユーザインタフェース上の理由である。たとえば「武市先生は」(たけいちせんせい) を得たい場合、ユーザは「421430316」を入力する。これに対して単純な式 (6) の値からは最適候補として「竹内先生は」、第 2 候補は「竹内新政府は」となる。しかし、我々のシステムでは第 1 候補の第 1 単語「竹内」を同じ長さの別の単語で置き換えた「武内先生は」「武市先生は」「高市先生は」を得て、これを式 (6) の順に整列し提示する。ここで、ユーザが「武市」を選ぶと、第 2 の単語「先生は」に対して、同様の処理が行われる。

候補が表示されたら、ユーザは次のいずれかのコマンドを用いて候補を選択する。

- 候補の選択
- 第 1 単語の切れ目を補正して再探索
- 次候補を表示

これらのコマンドは携帯電話の上下左右キーと決定キーを用いた操作に容易に置き換えることができる。

本システムを使用する際には、ユーザが候補を確定するたびに学習コーパスを更新してオンライン学習を行い、日本語入力をユーザ入力に対して最適化する。

#### 4.4 TouchMeKey による入力例

TouchMeKey のうち特に CWC による入力例を図 1 に示す。右側のボタン群の左 3 列に 10 のキーが示され、最右 1 列に選択のためのキーが示されている。ユーザが入力した数字は画面中ほどの横に細い部分に



図1 TouchMeKeyによる入力例

Fig.1 An input example using TouchMeKey.

示され、単語候補は下部分に、入力結果は上部分に表示される。ユーザはこのソフトウェアを用いてマウスやタッチパネルを用いて入力を行うことができる。このソフトウェアでは濁点は入力する必要はなく、たとえば「にほんご」と入力するときには「5602」とボタンを押す。そのため\*ボタンは備えておらず、句読点や記号は0ボタンに割り当てられている。

例文は「武市先生はいつも忙しい」である。ここでは、第3著者の電子メールを1万単語学習して入力が行われている。「武市」「先生」「忙しい」などの入力においては補完の効果がうかがえる。この入力例からみても、かな漢字変換の7倍もの候補を処理している

ようには感じられないだろう。我々の言語モデルの有効性を体感できる。

## 5. 評価実験

### 5.1 学習を用いない場合

まず、PPMによる学習を行わず、隠れマルコフモデルだけを用いる場合の入力効率を評価する。従来方式は補完を行わないものが多いため、CPについて、KPと比較する。

ここでの評価は、文書を入力する際の単語あたりの平均打鍵数により行う。テスト文書としては、経済面記事、および第3著者の電子メールを用いた。これら

表 3 経済面の記事の単語あたり打鍵数

Table 3 Number of the keystrokes per word for articles of the economic domain.

	数字列 入力	単語切れ 目補正	選択
CP	2.866	0.116	1.903
かな方式	8.088	0.013	1.088
ボケベル方式	5.731	0.013	1.088

表 4 1 個人の電子メールの単語あたり打鍵数

Table 4 Number of the keystrokes per word for an personal e-mail text.

	数字列 入力	単語切れ 目補正	選択
CP	2.326	0.131	2.065
かな方式	6.522	0.016	1.080
ボケベル方式	4.651	0.016	1.080

2 文書に対して、子音変換、かな方式の文節変換、ボケベル方式の文節変換でテスト文書を自動入力し、1 単語あたりの平均キーストローク数を計測した。

かな方式、ボケベル方式の自動入力における候補絞り込みは 3.1 節と同じものを用いている。打鍵数は、入力、切れ目補正、候補選択にかかった回数をそれぞれ示す。候補の  $n$  番目にあがった場合のキー入力数は  $n$  回であるものとした。

結果を表 3、表 4 に示す。まず、当然の性質として、数字列入力においてはボケベル方式は子音漢字変換の 2 倍の打鍵数がかかる。また単語切れ目変更および選択の数は、かな方式とボケベル方式は同数となる。

子音漢字変換では、従来の手法に比べ、数字列入力の打鍵数が激減している。その分、選択に要する打鍵数が増えている。選択においては、かな漢字変換では、ほとんどの目的の単語が第 1 候補にあがっていたのが、子音漢字変換では、平均的には第 2 候補になってしまったことが分かる。とはいえ、7 倍もの候補を扱っているのに対し、この程度の増加で済んでいることから、隠れマルコフによる言語モデルの有効性が検証されている。

選択のための 1 打鍵と数字列入力のための 1 打鍵では時間のかかり具合が異なるため、単純な和には疑問が残る。とはいえ、総合的な労力の 1 つの指標にはすることはできるであろう。総和でみると、かな方式に対しては平均 39%、PB 方式に対しては平均 15%も

CP において 1 単語あたりの数字列の入力数が 2.2 節の 4.81 よりも少ない。これは、2.2 節のものは、異なり単語の平均長であるのに対し、ここでは単語の重複を許したテストコーパスを用いた入力実験であるからである。

表 5 単語単位の補完なし入力における 1 段階方式と 2 段階方式の比較

Table 5 Comparison of the number of the keystrokes (our method vs. Tegic 9's method).

	打鍵数	経済面記事	電子メール	教科書
1 段階 変換	子音かな変換時 total	1.783 4.649	1.959 4.285	2.599 4.845
2 段階 変換	子音かな変換時 かな漢字変換時 total	1.445 0.621 4.933	1.567 0.385 4.278	1.717 0.510 4.473

打鍵数数を減らすことができることが分かった。

## 5.2 既存の類似システムとの比較

本節では Tegic 9<sup>1)</sup> や ZI<sup>9)</sup> の用いている 2 段階変換システムとの効率の比較を行う。Tegic 9 や ZI のシステムでは、ユーザは子音を入力し、これをかなに変換し、さらにはかな漢字変換を行う。Tegic 9 や ZI はそのシステムの詳細を公開していない。そこで、最少の打鍵数で済む 2 段階方式を工夫して実装した。比較は最も単純な CW に対して行った。

表 5 にその結果を示す。変換時の打鍵数とは目的の単語が平均何番目に現れたかを表す。特に 2 段階方式では子音かな変換とかな漢字変換とそれぞれにかかった選択時の打鍵数が記載されている。このとき、かな漢字変換時の打鍵数は、総単語数で割っているため、1.0 よりも小さくなっている（つまり、かなで記述する助詞などの入力はかな漢字変換は行われぬが、その分も単語数に含めて 2 段階目の変換の平均打鍵数を出しているためである）。total はこれに単語あたりの入力のための打鍵数（単語中の子音の平均数）を足した値を示している。

2 段階方式ではやはり選択にかかる打鍵数はそれぞれをみると 1 段階方式よりも小さい。特にその傾向が教科書の場合にはよく現れている。他の 2 つの文書に関しては、1 段階も 2 段階も結果は差がないことが分かる。であれば、変換の手間のより少ない 1 段階方式の方がユーザにとって良いのではないかと我々は考えている。

## 5.3 学習を用いる場合

同様の自動入力実験をユーザコーパスによる学習を用いたうえで行う。図 2、図 3、図 4 がユーザコーパスを変えた場合の結果を示している。

グラフを作成するにあたっては、オンライン学習の機能は外した。それは、ある一定量のユーザコーパスを用いたときの平均的な打鍵数を計測するためである。横軸はユーザコーパスの大きさ、縦軸は各学習コーパスを用いたときの単語あたりの総打鍵数を表している。

各グラフ中に 4 本の線があるが、この線は上から順

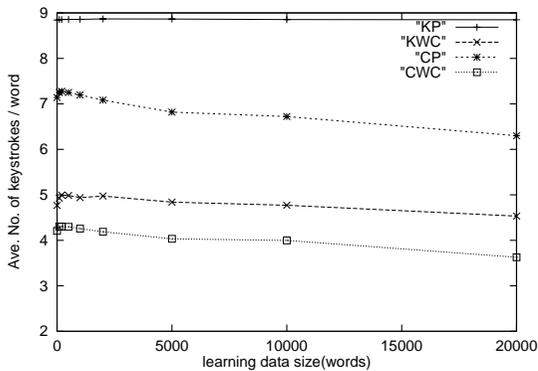


図 2 経済面記事での学習効果

Fig. 2 Learning effect for articles of the economic domain.

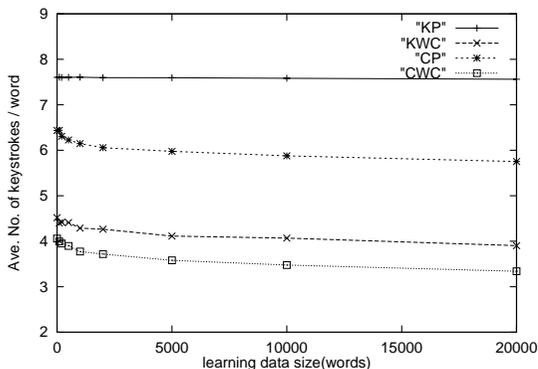


図 3 電子メールでの学習効果

Fig. 3 Learning effect for an personal e-mail text.

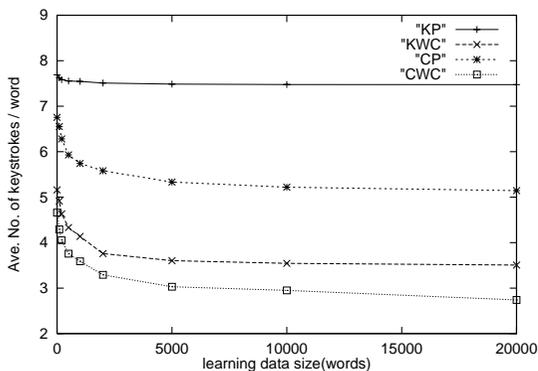


図 4 教科書での学習効果

Fig. 4 Learning effect for a programming language textbook.

に、KP、KWC、CP、CWCを表している。補完を用いた入力(KWCとCWC)の際の単語選択には曖昧性がある。これに関しては、入力の途中で第1候補にあがった場合はその段階で単語を選択し、第1

候補にあがらなければ音入力を続行するものとした。単語を最後まで入力して、目的の語が第1候補にあがらなければその単語を選択し、そのときの選択のための打鍵数は前節と同様である。

まず、ユーザコーパスの大きが増えると、1単語あたりの打鍵数が減っている様子がうかがえる。この傾向はKP以外の3方式いずれにもあてはまる。また、学習の度合いは、CWCのときに最も大きい。したがって、学習は入力的方式に曖昧性があるほど効果が大きいことがうかがえる。

KP以外の3方式に関しては、10,000単語ですでに学習効果が現れている。学習効果の得られる語数が少なくても済むのは、学習効果が短期間で得られるうえ、携帯電話のように小容量の機器類上にユーザコーパスを持つことができるため、重要である。

さらにどこまで打鍵数が減るかを調べるため、CWCを用いてユーザコーパスを増やしてみた。具体的には経済面、メールについては15万単語、教科書については5万単語である。すると、単語あたり必要なキー入力数は経済面記事の場合に3.31、電子メールでは3.07、教科書では2.70となり、それぞれ21%、24%、42%も削減されることが分かった。つまり、打鍵数はCWCを用いて学習を行うと、現行のかな方式に比べ半数を大きく下回る打鍵数で済むという劇的な効果があることが分かった。

3つのグラフを比べると、教科書が最も学習効果が高い。これは、教科書の語彙が3つの中で最も少ないためである。一般にユーザコーパスは、新聞記事などと比べ異なり語彙数が少ないが、このような文書は、学習に向いている文書であることが分かる。

#### 5.4 ユーザ実験

最後に、ユーザによる入力実験の結果を示す。4.4節に示したTouchMeKeyと図5に示した入力用装置を用いて、入力実験を行った。このときの学習コーパスの量は1万単語である。被験者は合計10人で、5人がKWC、残りの5人がCWCの方式を用いて入力を行った。各被験者の携帯電話におけるメールの普段の利用状況を表6に示す。文章は学習コーパスに存在する単語のみからなる、学習コーパスには含まれない文を入力するものとした。入力は30分を1セッションとし、10セッション、計5時間の入力実験を行った。セッション中は入力だけに専念するものとした。2つのセッション間の休み時間は指定せず、日をまたがってもよいものとした。

図6に速度の変化を示す。横軸はセッション、縦軸にそのセッション中の平均入力速度をプロットした。



図 5 入力に用いたデバイス

Fig. 5 Input device used for the user evaluation.

表 6 各被験者の携帯電話におけるメールの利用状況

Table 6 How often the subjects use mobile phone e-mails.

被験者	利用頻度
KWC-1	1 週間に数回
KWC-2	1 週間に数回
KWC-3	1 か月に数回
KWC-4	ほとんど使わない
KWC-5	ほぼ毎日
CWC-1	1 か月に数回
CWC-2	ほぼ毎日
CWC-3	ほぼ毎日
CWC-4	ほぼ毎日
CWC-5	持っていない

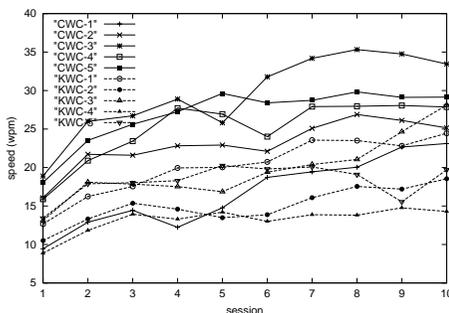


図 6 10 人のユーザによる入力効率の変遷

Fig. 6 Speed change of the 10 subjects.

速度は、1 分あたりの入力単語数 (wpm) で計測した。10 本の折れ線がそれぞれ 10 人の被験者の入力速度の変化を表している。実線は CWC、点線は KWC による入力を表す。

CWC については、1 セッション目では被験者の入力速度が 9 wpm から 19 wpm の間に分布しているが、5 時間後の 10 セッション目にはそれが 23 wpm から

33 wpm になっている。各被験者ごと 1 人ずつみても、速度が 10 wpm 以上上がっている。一方 KWC では、1 セッション目の速度は 9 wpm から 13 wpm の間に分布しているが、10 セッション目では 14 wpm から 28 wpm になり、分布が広がっている。つまり、入力速度が速くなる人と速くならない人がいた。CWC は KWC より人間による学習が効果を奏することが分かる。10 セッション目の 5 人の被験者の速度の平均をとると、KWC が 21 wpm であるのに対し CWC は 27.7 wpm であり、CWC の方が約 3 割速い。普段携帯電話でメールを利用していない CWC-1 や CWC-5 も 5 時間後には 23 wpm, 29 wpm といった KWC の平均を上回る速度で入力できるようになっている。

最高速度をみると、CWC-3 が 8 セッション目に 35 wpm を記録している。わずかに 10 のキーによる入力であることを考慮すると驚くべき結果である。

## 6. 結 論

本稿では携帯電話のための日本語入力方法、特に子音漢字変換による入力を検討した。子音漢字変換では、ユーザは子音を入力し、これを子音漢字変換ソフトウェアが直接漢字混じり文書に変換して提示し、ユーザが候補中から目的のものを選択することにより文書の入力が行われる。

従来の方式では、ユーザはかなを入力するため、1 かなあたり複数回のキー入力が必要であった。一方で子音入力方式では、1 かなあたり 1 回の入力で済む。しかし、その分子音漢字変換はかな漢字変換に比べ、じつに 7 倍もの単語候補を扱う必要が出てきてしまう。本稿ではこの大量の候補をユーザの単語選択に適合する順序で整列させることにより解決を試みた。具体的には、隠れマルコフモデルに PPM を融合させることにより、動的な言語モデルを提案して用いた。

実際に子音漢字変換ソフトウェア TouchMeKey を実装を行ってさまざまな評価を行った。第 1 に子音漢字変換自体の効率を計測したところ、入力文書の語彙が個人のメールなどに限定された場合には 4 割程度も打鍵数を削減することができることが明らかとなった。第 2 に学習効果を検討したところ、10,000 単語の学習でも効果が得られ、学習コーパスを増やすにつれ、打鍵数が従来方式に比べ半減するという大きな効果が得られることが分かった。最後にユーザ実験を行ったところ、熟練したユーザは十分高速に本システムでの文書入力が可能であることが示された。

今後の課題としては、連文節を用いた補完の方式を検討することがあげられる。また、実際に携帯電話上

で採用されるべく、努力したい。

### 参 考 文 献

- 1) Tegic 9. Tegic 9 home page (2000).  
<http://www.t9.com>
- 2) Ppm 法を用いたかな漢字変換の学習モデル, 情報処理学会自然言語処理研究会論文集, Vol.146 (2001).
- 3) Bell, T.C., Cleary, J.G. and Witten, I.H.: *Text Compression*, Prentice Hall (1990).
- 4) 携帯電話の送りすぎで親指障害? (2001).  
<http://www.hotwired.co.jp/news/news/20010530304.html>
- 5) Manber, U. and Myers, G.: Suffix arrays: A new method for on-line string searches, *SIAM J. Comput.* (1993).
- 6) NTT 西日本: 案内ジョーズのホームページ (2000). <http://www.ntt-east.co.jp>
- 7) W.J., et al.: Teahan: Probability estimation for ppm, *NZCSRSC'95* (2000). <http://www.cs.waikato.ac.nz/wjt/papers/NZCSRSC.ps.gz>
- 8) Ward, D., Blackwell, A.F. and MacKay, D.J.C.: Dasher: A data entry interface using continuous gestures and language models, *ACM Symposium on User Interface Software and Technology* (2000).
- 9) ZI-Corp: Zi home page (2000). Available from <http://207.229.18.241/>
- 10) 永田昌明: 確率モデルによる日本語処理に関する研究, 博士論文 (1998).
- 11) 増井俊之, 小松浩之, 高林 哲: 日本語動的単語補完方式 nanashiki を活用した予測入力, *WISS* (2001).
- 12) 松本裕治ほか: 日本語形態素解析システム「茶筌」使用説明書, 奈良先端科学技術大学院大学 Technical Report (1997).
- 13) 東京芝浦電気(株): 日本語入力装置, 特開昭 57-185528 (1982).

(平成 13 年 11 月 12 日受付)

(平成 14 年 9 月 5 日採録)



田中久美子 (正会員)

1997 年電子技術総合研究所勤務を経て, 2000 年より東京大学大学院情報学環講師, 現在に至る。博士 (工学)。計算言語学, 自然言語処理の研究・教育, プログラミング言語の教育に従事。近年は特に言語の実時間動的処理に興味を持つ。日本ソフトウェア科学会, 言語処理学会, ACL 各会員。



犬塚 祐介 (正会員)

1977 年生。2000 年東京大学工学部計数工学科卒業。2002 年東京大学大学院工学系研究科情報工学専攻修士課程修了。ユーザインタフェース, 自然言語処理に興味を持つ。現在 NEC ネットワークスモバイルターミナル事業本部勤務。



武市 正人 (正会員)

1948 年生。1972 年東京大学工学部助手, 講師, 電気通信大学講師, 助教授, 東京大学工学部助教授を経て 1993 年東京大学大学院工学系研究科教授 (情報処理工学講座), 2001 年より同大学大学院情報理工学系研究科教授, 現在に至る。工学博士。プログラミング言語, 関数プログラミング, 言語処理システムの研究・教育に従事。日本ソフトウェア科学会, 日本応用数理学会, ACM 各会員。