

携帯電話の10keyを用いた日本語入力 —子音だけで日本語が入力できるか—

田中(石井)久美子

犬塚祐介

武市正人

東京大学

1 はじめに

近年では携帯電話が普及し、さらにi-modeなどの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10keyを用いて入力する。10keyを用いた日本語入力には、かな方式やポケベル方式などが挙げられるが、これらは、キーを押す回数が多くなってしまい、という問題点がある。

そこで、本稿では、キーを押す回数を減らした日本語入力方式を提案する。通常の計算機上での日本語入力は、子音と母音を交互に入力して、かなを決定してから漢字へ変換する。一方で本稿で提案する方式は、子音に相当する数字のみを入力し、目的とする漢字へ変換する。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」と入力して目的とする語を推定する。10keyでは、nが5、hが6、nが0(わ行)、gが2#に対応し、「5602#」を入力する。

そもそも、子音だけ用いた入力は、現在の日本語入力が発達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[10][11]。しかし、その後、パソコンでの入力がローマ字入力等へと確立されるにしたがって、子音だけを用いた入力方式の研究はすたれたようである。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、[4][7]などは英語に対して同様のシステムを提案しているほか、日本語入力の実装にも着手しているようである。また、NTTの案内ジョーズ[2]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。

一般に、子音を用いた入力は、従来のかなを入力する方法に比べ、挙げられる語候補が増大するため、そもそも実現が可能であるかどうか問題となる。本稿では、まずこの実現可能性について毎日新聞のテキストデータを解析して論じた上で、本入力方式を実装した結果を報告する。最後に本システムを用いてテキストを自動入力し、従来の入力方式と比較検討する。

2 入力方式

従来の10keyを用いた入力の方式は、いずれも「かな」を入力する。したがって、子音と母音の両方を数字により決定しなければならない。子音には、一般に表1のように数字を割り当てる。たとえば、「は」行のかな文字を入力するには、6を使う。一方で、母音には「あいうえお」にそれぞれ12345を対応させる。そして、この母音に対応する数字をどのようにユーザが入力するかにより、二つの方式がある。

第一の方式は、「かな方式」といわれ、キーを押す回数により母音の番号を入力する。たとえば、「は」は子音6番の母音5番であるので、6を5回押すこととなる。この方式は、ユーザの指の移動が少ない、という長所があ

るものの、さまざまな問題がある。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。さらに同じかな文字を連続して入力する場合に問題がおきる。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、「ひ」を入れた後に、別のキーを押して、文字を確定せざるを得ない。したがって、連続入力の時だけ入力方式が異なり、ユーザは一貫した入力を行うことができない。

第二の方式は、「ポケベル方式」(以下PB方式と記述)といわれ、この方式では母音の番号のキーを直接押す。たとえば、「ほ」であれば、6の後に5のキーを押す。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、キーを打つ位置が常に移動するため、携帯電話上の小さな10key上での指の移動は、やはり誤入力を引き起こす¹。

以上の問題点を整理すると、携帯電話での入力においては、指の移動を減らし、さらにキー操作の回数を減らすことが重要となることがわかる。

本稿では、キー操作回数を減少させるために、ユーザは母音を入力せず子音だけを入力することにより、日本語テキストを入力する方式を提案する。この方式では、「にほんご」を入力する場合は、図2のように従来方式よりもキー操作数が少ない。

このような入力においては、キーを押す回数は著しく減少するが、一方で入力に対応する候補が多くなる。たとえば、「にほんご」を表す「5602#」は「にほんが」をも表す。つまり、本方式を用いた日本語入力システムを用いる際の問題点は、通常のかな入力よりも単語候補が増えるため、これをどのように扱うか、という点に集約される。

3 実現可能性

まず、本方式を用いた入力には、一段階入力と、二段階入力の二つが考えられる。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。一方の二段階入力は、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、後者は処理が二段階になるという問題が挙げられる。[4]や[7]は後者の方式を用いている。しかし、ユーザの観点からは候補を適切に処理可能であるのであれば、一段階入力が望ましいであろう。そこで、まず候補数について検証する。

¹ いずれの方法も濁音の入力には、#キーを用い、濁音は母音の1、半濁音は母音の2と割り当てられている。たとえば、「べ」の入力には、「へ」を入力した後、#を押すが、「かな入力」の場合には、#を2回押すのに対し、ポケベル方式では#の後に2を押す。また、「を」や「ん」は、それぞれ「わ」行の母音2と3と定義されている。

表 1: 10key 上へ子音の割り当て

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)

表 2: 各方式で「にはんご」を入力する時のキー操作

	に	ほ	ん	ご
かな方式	55	66666	000	22222#
PB方式	52	65	03	25#
本方式	5	6	0	2#

たとえば、毎日新聞(94年)の社会面(総単語数 130 万語であり、約 28Mbytes)には異なり語数で約 9 万の単語がある。これらを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 50 のかな表現があるので、 $50^{4.81}$ = 約 1 億 4290 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、その分だけしか使っていないことになる。すなわち、かなによる入力も、もともと非常に冗長であり、社会面程度の単語数であれば、同音異義の候補がないと仮定した場合には、一つの入力に対して、候補は 1 以下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、 $10^{4.81}$ = 約 63100 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。

さらに、実際の候補数という観点でこの問題を論じる。一単語のユーザ入力に対して、漢字変換まで含んだ単語がいくつ対応するかを換算してみる。すると、表 3 の 1 行目に示すように、本方式では単語候補は 1 単語につき 2 語を越えてしまう。一方でかな入力の場合には、2 単語を大きく下回っている。また、最大値も、著しく大きい。これは単漢字の場合の同音異義語の候補の多さを表している。最大の 167 となった数字列は「21」であり、候補は「解」「貝」「回」「慶」「鯉」「糸」などとなる。

この値は、使用頻度を考慮しない値となっているが、一般に言語とは、よく使う単語ほど曖昧である。そこで、頻度分布にしたがって任意の単語を選択し、入力した時に平均的に挙がる候補数を計算してみた結果が 3 行目である。従来方式と比べて、著しく大きな値であり、本方式を用いた日本語入力は一見非現実的のように思える。

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 31.6 単語であったとしても、それらを適切に並べて提示することができれば、本方式も実用に耐えうる。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、必ずこの 2 単語を上位に提示すればよい。そこで、頻度 100 以上を頻出語として統計を取り直した結果を表 3 の 4 行目から 6 行目に示した。頻度を考慮しても平均単語候補数は一気に押さえられる。

また、品詞を利用することも考えられる。そこで、単語を品詞別に扱い、同様の統計を計算した結果を 7 行目から 9 行目に示した。現実的な統計結果が示されている。

また、携帯電話は本来私用のための機器であるから、辞書をユーザに特化することも考えられる。そこで、第三者者の 1 年分の電子メール(総語数が 168258 語、異なり語数は 8427 語)を用いて統計結果を得た。こちらも大きく平均単語候補数が減少したことがわかる。

以上から、1. 頻度、2. 品詞、3. ユーザコーパスの 3 つを利用することにより候補を適切な順序に並べ替えることができると予想され、三つを組み合わせるシステムを設計すると、従来方式からさほど劣ることなく本方式で

表 3: 1 単語あたりの平均候補数

		本方式	従来方式
社会面	平均単語候補数	2.41 単語	1.39 単語
	最大単語候補数	167 単語	43 単語
	頻度を考慮した平均単語候補数	31.61 語	4.76 単語
類社 出会 語面	平均単語候補数	1.82 単語	1.15 単語
	最大単語候補数	32 単語	10 単語
	頻度を考慮した平均単語候補数	6.29 語	1.75 単語
品社 詞会 別面	平均単語候補数	1.69 単語	1.19 単語
	最大単語候補数	74 単語	37 単語
	頻度を考慮した平均単語候補数	5.47 語	1.65 単語
ユーザ	平均単語候補数	1.69 単語	1.14 単語
	最大単語候補数	51 単語	15 単語
	頻度を考慮した平均単語候補数	12.50 単語	3.02 単語

も日本語の入力が実現される可能性がある。そこで、次節では候補絞込のための言語モデルと学習モデルを説明する。

4 統計モデル

4.1 言語モデル

本稿では、候補を確率モデルを基礎として並べ替えることにした。ユーザが入力した数字列を C 、単語列を W とすると、最適な単語列は、

$$\hat{W} = \arg \max_W P(W|C) \quad (1)$$

と定義できる。ここでまず、ある C が与えられた時に候補となるすべての W に共通な文字列であるので、

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

である。すなわち、候補の中から最適な単語の並びを求めることとなる。本稿では品詞情報を用いるので、 T を品詞の並びとすると、 $P(W)$ は一般性を失わずに

$$P(W) = \sum_T P(W|T) \quad (3)$$

ここで、

$$P(w_n|w_{1,n-1}, t_{1,n}) = P(w_n|t_n) \quad (4)$$

$$P(t_n|w_{1,n-1}, t_{1,n-1}) = P(t_n|t_{n-1}) \quad (5)$$

と近似すると、式 3 の右辺は、

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

と変形できる。これは品詞の隠れマルコフモデルとなっており、仮名漢字変換で用いられることの多い接続コストを最少とするモデルに統計的意味付けを行ったものである [8]。すなわち、候補は上記確率の値の順にユーザに提示されることとなる。

4.2 学習モデル

本稿ではユーザコーパスを用いて候補の提示順序をユーザに特化する。ここで問題となるのは、ユーザの

コーパスは、一般に量が小さいということである。とすると、上記のモデルでは、さまざまな単語に対して、有意な統計をとれないおそれがある。そこで、PPMを用いて式(6)単語確率をユーザコーパスに特化する。

Witten[5]らは、ある時系列 $S=s_0 \dots s_i$ が与えられた時、つぎの要素 s_{i+1} の確率を次式で推定する。この方法は blending と呼ばれている。

$$P(s_{i+1}) = \sum_{o=-1}^{omax} u_o P_o(s_{i+1}) \quad (7)$$

ここで、 o とは s_{i+1} の前 o 単語のことであり²、特に $o = -1$ の時は、文脈から情報が全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。 $P_o(s_{i+1})$ は前 n 単語を考慮した確率で、 C_o をその文脈の現れた回数、 $c_o(s_{i+1})$ をその文脈で s_{i+1} が現れた回数として、

$$P_o(s_{i+1}) = \frac{c_o(s_{i+1})}{C_o} \quad (8)$$

で近似される。

u_o は直前の o 単語を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 [6]。本稿の問題の場合には、直前に採用された単語候補が、その頻度が小さかろうと重要となると思われるので、PPMA[5]を用いるものとした。すなわち、

$$u_o = \left(1 - \frac{1}{C_o + 1}\right) \times \prod_{i=o+1}^{omax} \frac{1}{C_i + 1} \quad (9)$$

また、 o の上限 $omax$ は同研究の成果 [6] をそのまま採用し 4(5-gram) とした。

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。それを通して文脈が反映される確率補正が十分研究され、有意な結果が得られている。さらに、わずか数語といった非常に小さなコーパスからも確率の補正が可能である点で携帯電話上のテキスト入力ユーザ特化には向いていると考えられる。また、PPM は本入力システム以前にも [3] や [1] など、ユーザインターフェース分野のテキスト入力の研究において用いられ、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。

5 実装

ユーザの入力方式にはかな方式、数字方式、以外にも、単語入力、文節入力、さらに補完の採否などさまざまなものが考えられる。ここで、補完とは、単語の途中までの入力、その単語を終わりまで推測するものである。本稿では、補完を用いる単語変換と、補完を用いない文節変換を実装することにした。

まず、辞書について説明する。 $o=-1$ の場合の単語辞書は、毎日新聞の社会面から構築した。その理由は、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。また、 $o=0$ 以上の場合の辞書は、ユーザコーパスから作成する。ユーザコーパスは、§3 で述べた第三者者の 1 年分の電子メールを用いる。コーパスを形態素解析し [9]、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。単語辞書には、単語、よみ、数字列、品詞、頻度

²通常自然言語処理で言われる n -gram は、PPM では、 $o=(n-1)$ となる。たとえば、 $o=1$ の時は、自然言語処理における bigram を、 $o=2$ の時は、自然言語処理における trigram を意味する。

が記述されている。また、 $o > 0$ の場合の文脈辞書の作成に際しては、コーパスの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て使用した。

これらを用いてユーザの数字列をかな漢字に変換する。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、まず式(6)が最大となる候補を永田 [8] の方式を用いて探索する。第二候補以下は、最初の単語と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せねばならず、使い勝手がよくない、というユーザインター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)を入力したい場合、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」、第二候補は「竹内新政府は」となる。しかし、システムは第二候補以下はそもそも探索もせず、第一候補の「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、これを確率順に整列し提示する。ここで、ユーザが「武市」を選ぶと、「先生は」に対して、同様の処理が行われる。

候補を表示されたら、以下のコマンドを用いてユーザは候補を選択する。

- #n 候補 n 番目の第一単語を選択
- a 最適候補を一括選択
- s 第一単語の切れ目を 1 文字短くして再探索
- l 第一単語の切れ目を 1 文字長くして再探索
- m 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。

実際に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辞書を更新して学習を行い、日本語入力をユーザに対して最適化する。また、辞書に登録されていない単語、すなわち未知語の場合には、ユーザは一字づつ入力をし、これを単語として登録する。しかし、本稿では、あるユーザコーパスの大きさに対する効率を定量的に測定したいため、実験に際してはこれらのオンライン学習は行わない。

6 評価実験

6.1 学習を用いない場合

まず、文節変換の場合において、従来方式と本方式を比較する。文書を入力する際に各単語あたり平均何回キーを押したかにより、評価を行う。テスト文書は、社会面記事、経済面記事 2 種類を用意した。社会面は辞書を作成するデータに含まれるものであり、経済面は辞書を作成するデータとしては用いられていないという差がある。

これら文書に対して、本方式、かな方式、PB 方式でテスト文書を自動入力し、それぞれ何回キー入力が必要となるかを計測した。テスト文書の未知語はすべて除き、いずれの入力方式においても既知であった単語について一単語あたりの平均キーストローク数を計測した。このような語は社会面記事 960 単語、経済面記事 830 単語であった。かな方式、PB 方式の自動入力における候補しぼり込みは 4.1 節と同じ方式を用いている。キー入力数は、入力、切れ目補正、候補選択にかかった回数の総和とした。候補の n 番目に挙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 4 に示す。かな方式に対しては平均 39%、PB 方式に対しても平均 15% もキー入力回数を減らすことが

表 4: 学習を用いない場合の平均キー入力回数

	社会面	経済面
本方式	4.14	4.12
かな方式	6.53	7.04
PB 方式	4.65	5.05

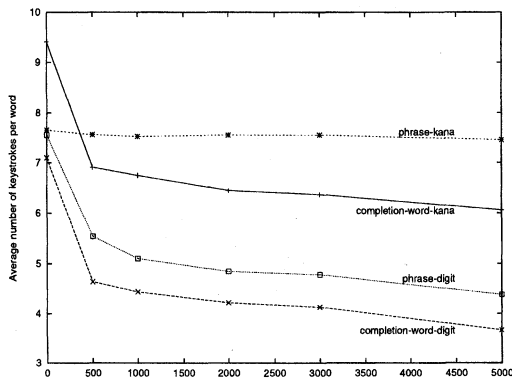


図 1: 学習ユーザコーパスとキー入力数

できることがわかった。すなわち、本方式は現状のかなを基本とする入力方式に劣るどころか、勝る。

6.2 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上で行う。線は上から順に、文節変換でかな入力、補完を用いた単語変換でかな入力方式、文節変換で数字入力補完を用いた単語変換で数字入力の場合をそれぞれ表している。補完を用いた単語変換の場合は、数字入力の途中で第一候補に挙げた場合はその段階で単語を選択し、第一候補に挙げられなければ単語の数字入力を続行するものとした。単語を最後まで入力して、目的の語が第一候補に挙げられなければその単語を選択し、その時の候補選択数は前節と同様である。

入力のためのテスト文書には、学習に用いていないユーザコーパスの一部を用いる。上記の実験同様、入力した単語のうち、全入力実験において未知語ではない 607 単語に対して単語あたりの平均キー入力数を計測し、プロットした。学習に用いたユーザコーパスは、少量のものは大容量のものを含む。

結果を図 1 に示す。まず左端の学習を用いない場合の線の始点の位置関係を見ると、ユーザコーパスをテスト文書とした場合においても、本方式の方がかな方式よりも一単語あたりの入力数が少ない。特に補完を用いた単語入力方式を用いると、その差は大きい。そして、補完を用いた単語変換で数字入力の場合は常に他の全方式に勝っていることがわかる。

第二に、ユーザコーパスの大きさが増えると、一単語あたりの入力回数が着実に減っている。この傾向は文節変換を用いたかな方式以外にも当てはまることから、学習は入力が曖昧な場合に効果が大きいということがいえる。

第三にユーザのコーパスがわずか 1000 単語ほどでも、かなりの学習効果が挙がるということがわかる。携帯電話はメモリに厳しい制限があることや、ユーザのコーパスをオンラインで集めることを考えると、1000 単語で効果が挙がるという結果は重要である。実際、補完を用いた単語入力の場合には、1000 単語の時に 38% のキー入力数が削減されている。

さらにどこまでユーザ入力が効率化されるかを調べるため、ユーザコーパスを 15 万単語にまで増やしてみた。すると、学習を用いない場合と比較して 55% のキー入力数が削減されることがわかった。この時に単語あたり

必要なキー入力数は 3.54 であった。

これほど学習に効果がある理由の一つとして、学習対象がユーザのメールのコーパスであることが挙げられる。ユーザの電子メールのコーパスは、総単語数が同じ文書と比較した時に、異なり語彙数が少なく、表現に限界がある。つまり、ユーザコーパスとは、そもそも学習には向いているものなのである。実際、同様の実験を経済面のコーパスを学習コーパスとして用いてみたところ、15 万単語で、38% の削減であった。とはいえ、55% という結果はあくまでユーザのコーパスに対して得られた値であるので、今後は他のユーザコーパスに対する実験を急務としたい。

7 結論

本稿では携帯電話における 10key を用いた日本語テキスト入力方式として、子音列の入力から目的とする日本語を推定し、ユーザがそれを選択する方式を提案した。本方式では、子音だけを入力するので入力キー操作回数を劇的に減らすことができる一方で、入力がより曖昧であるので語の候補数が増えてしまう。本稿ではまず候補の適当な整列手法について毎日新聞のコーパスを用いて検証した。これによると、1. 頻度 2. 品詞 3. ユーザコーパスの三つが鍵となることがわかった。そこで、これに基づいて日本語入力システムを HMM を用いた言語モデルと PPM を用いた学習モデルを融合させて設計し、実装した。本システムでは数字入力に対してかな漢字混じりの最終的な日本語を出力するもので、補完を行う単語変換、および、補完を行わない文節変換の二種類の入力が可能である。

このシステムを用いて 500 単語以上のテスト文書を自動入力し、キー操作の回数を検証すると、現在最も多く用いられている携帯電話における日本語入力方式に対して 39% もキー入力数が削減されることが明らかとなった。さらに、著者のユーザコーパスを用いて学習したところ、1000 単語の学習で 38%、15 万単語で 55% のキー入力数が削減されることが明らかとなった。

参考文献

- [1] Daniel Ward et. al. Dasher: a data entry interface using continuous gestures and language models. In *the ACM Symposium on User Interface Software and Technology*, pages pp.129-137, 2000.
- [2] NTT 西日本. 案内ジョーズのホームページ, 2000. <http://www.ntt-west.co.jp>.
- [3] Masui T. Po_box an efficient text input method for handheld and ubiquitous computers. In *the ACM Symposium on User Interface Software and Technology*, pages pp.113-119, 1999.
- [4] T9. Tegic 9 home page, 2000. <http://www.t9.com>.
- [5] Bell T.C., Cleary J.G., and Bell I. H. *Text Compression*. Prentice Hall, 1990.
- [6] W.J. Teahan. Probability estimation for ppm. In *NZCSRSC'95*, 2000. <http://www.cs.waikato.ac.nz/wjt/papers/NZCSRSC.ps.gz>.
- [7] Zi-Corp. Zi home page, 2000. Available from <http://207.229.18.241/>.
- [8] 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.
- [9] 松本裕治 他. 日本語形態素解析システム「茶筌」使用説明書, 1997. 奈良先端科学技術大学院大学 Technical Report.
- [10] 東京芝浦電気(株). 日本語入力装置. In 特開昭 57-185528, 1982.
- [11] 日本電気(株). 日本語入力装置. In 特開平 10-124506, 1996.