

## 少数キーを用いた日本語入力

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

本稿では文字入力用に4つだけのキーを用いて自由な日本語文章の入力を行うシステムとその効率を報告する。本システムは動的な言語モデルに基づいて予測的に入力を行うものである。応用としては、小型機器類における入力や、高齢者や障害者のための簡易文書入力システムが考えられる。さまざまな実験の結果、4つのキーを用いたとしても、学習を用いれば現行の携帯電話における文書入力に匹敵する効率で入力が可能であることが分かった。

### Entering Japanese with a Few Button Device

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

This paper presents the design of a text-entry device that requires only four buttons. Such a device is applicable as the text interface of portable machines and as an interface for disabled people. The text-entry system is predictive; the basis for this is an adaptive language model. Our evaluation showed that the system is at least as efficient for the entry of free text as the text-entry systems of current-generation mobile phones.

#### 1. はじめに

PDA や携帯電話、さらに時計型コンピュータにみられるように、近年は機器類の小型化が著しい。小型の機器類においては、ユーザが場所を問わず動的に情報を得ることができることが大きな利点となっている。しかし、その際に問題となるのが、ユーザの入力である。ユーザが情報を発信するためには、ユーザ自身が機器類を用いて文書入力を行う必要がある。対人の場合には、音声を用いることも可能であるが、そうでない場合にはユーザは電子的な文書入力を機器類上に装備されているインタフェースを用いて行わなければならない。

入力と言っても、スタイラスでストローク入力、キーによる入力などさまざまなものが考えられる。小型の機器類という制約下での入力では、キーによる入力が自然な入力方法の1つとしてあげられるが、フルキーボードの装備は大きめのタッチパネルがある場合など

に限定されてしまう。そこで、本稿では少数のキーを用いた文入力を論じる。

少数のキーだけで文書を入力する試みは、古くからある。代表的なものは速記用タイプライターである<sup>6)</sup>。ここでは、キー数が少ない分、複数のキーを同時に押すことにより、キー数よりも多い文字を表現する。速記用タイプライターを用いると、特殊訓練の後には話すスピードで入力が可能となる。一方で、90年代に入り、機器類の小型化が顕著となると、そのためのキーボードがさまざまに考案されるようになった。たとえば、片手用の Half Qwerty キーボード<sup>5)</sup> や、SHKey<sup>12)</sup> などがある。なかでも近年は、携帯電話上で約10個のキーによる入力が論じられている。英語においては James<sup>2)</sup> らがさまざまな入力方法の有効性の評価を行っており、また、日本語においては POBox<sup>4)</sup> や、カットキー<sup>11)</sup> が提案され、我々も携帯電話については「1かな1押し方式」について文献<sup>15)</sup>で別に有効性を論じている。さらにキーの数を減らす観点では、増井<sup>13)</sup>が本稿と類似のアイデアを先行して示している。ただし、その有効性や、現実的な入力を行うことのできるボタンの数に関する議論は行われていない。本稿の目的はこの点を論じ、合わせて言語に本質的に内在する冗長性に光を当てることにある。

少数キーによる入力の応用は携帯機器のほかにも考えられる。第1に、近年ではコピー機やファクシミリなど事務機器や家電までが通信機能を備えていること

<sup>†</sup> 東京大学大学院情報学環

Interfaculty Initiative in Information Science, The University of Tokyo

<sup>††</sup> NEC ネットワークスモバイルターミナルソフトウェア開発本部  
NEC Networks Mobile Terminals Software Development Division

<sup>†††</sup> 東京大学大学院情報理工学系研究科学

Graduate School of Information Science and Technology, The University of Tokyo

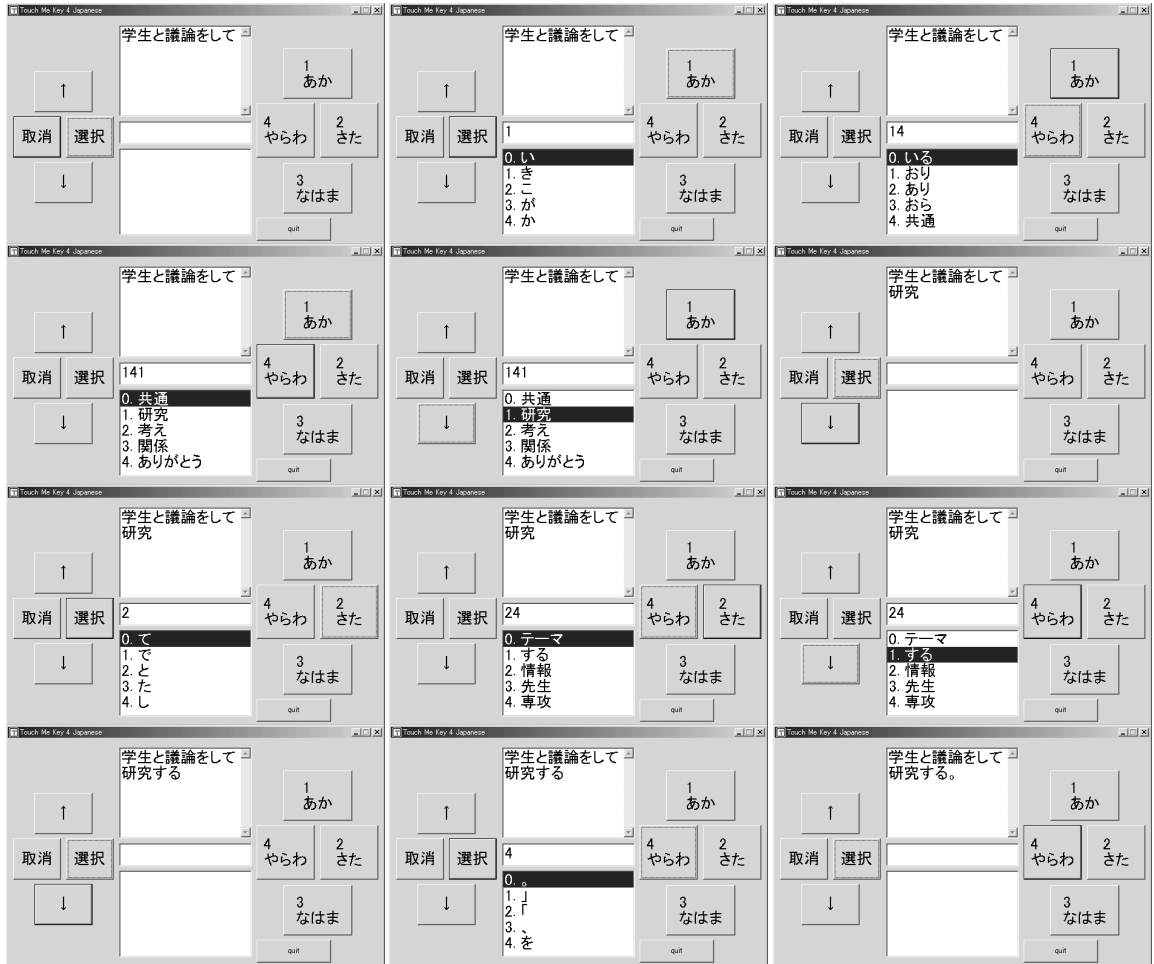


図 1 TouchMeKey4 による入力例

Fig.1 Example entry using TouchMekey4.

がある。通信のための手段として、フルキーボードを備えることも考えられるが、本稿で提案するような少数のキーを装備することにより簡易入力を実現することも一案となりうる。

第 2 の応用としては、高齢者や障害者のための簡易入力インタフェースである。社会の高齢化にともない、高齢者や障害者でも情報処理に関わる機会が増えつつある。その際の大きな問題の 1 つはキーボード操作であろう。そこで、たとえば 4 つのボタンを大きくタッチパネル上に表示して入力を行うと、高齢者向けの情報処理の道が開ける可能性がある。また、手などが不自由な身体障害者に関して、視線やタッチパネルを用いた入力の可能性が広がる。

以上のような可能性を見据え、本稿では少数キーを用いた日本語入力を論じる。まず、次章で我々が最終的に行き着いた 4 つのキーを用いた入力を示し、それ

以降の章で、この結果を得た理由を説明する。最後に有効性について実験結果とともに論じる。

## 2. 実 例

図 1 に我々の開発した TouchMeKey4 の GUI を示す。本稿で問題とする文字入力用のキーとは、文書入力の際に用いるキーだけであり、スクロール、取消、選択といった機能用のキーはすでに装備されているものとして考える。その理由は、機能用のキーは文書入力以外の目的にも必要とされるため、装備されていることが多いからである。TouchMeKey4 においては、入力用のキーとは右側に位置する 4 つのキーであり、左側に機能用のキーが配置されている。入力用の各キーにはかなが重複して割り当てられており、1 番キーに「あ」「か」「が」段、2 番キーに「さ」「た」「ざ」「だ」段、3 番キーに「な」「は」「ば」「ぱ」「ま」段、4 番

キーに「や」「ら」「わ」段および「ん」と記号が割り当てられている。

今すでに「学生と議論をして」が入力されており(図1の1行目左),これから続けて「研究する。」を入力するものとする。すでに入力済みの文書が表示されている部分をテキストボックスと呼ぶ。TouchMeKey4においては入力は単語単位で行う。また,ユーザが入力を行うときには,目的の単語のかなに相当するキーを1回ずつ押す。「研究」を入力するには「け」のボタン,すなわち1番ボタンを1度押す(図1の1行目中)。すると,TouchMeKey4の中ほどの横長のユーザ入力ボックスに,1が現れ,また中ほど下の大きな候補ボックスに複数の候補が出現する。TouchMeKey4は,長さ1の入力に対しては,長さ1の候補しかあげないため,候補ボックスにあがっている候補は「い」「います」の「い」「き」「きます」の「き」などとなる。候補ボックスには目的の単語「研究」は現れていないので,さらに入力を続ける。「け」の次の「ん」を入力するために4番ボタンを1度押す(図1の1行目右)。すると,候補として「いる」「おり」などがあがる。

ユーザ入力 that 長さ2以上になると,TouchMeKey4は長さ2以上で現状の入力に該当するあらゆる単語を候補とする。現在の入力の「14」に対してはほかに「考え」や「ありがとう」「関係」など大量の候補が挙がっている。これらの候補は,4章でより詳しく説明するが,簡単には大規模コーパス(新聞)中の出現頻度のほか,1万単語程度のユーザ文書中の統計を考慮して整列の上提示されている。特に,直前の単語列「と議論をして」がユーザ文書中にどのように現れたかをふまえたうえで候補は整列される。ここでのユーザ文書は第三著者の電子メール1万単語である。このため「いる」などの頻出語が上位の候補としてあがっているほか「おら」「あり」が上位にあがっている。これは,現行の文脈が「議論して」であり,さらに「しておらず」「してあり」などがユーザ文書の中にみられたためである。

現時点でも候補ボックスには「研究」はあがっていない。そこで,さらに「き」を入力するために1番ボタンを押す(図1の2行目左)。すると,今度は候補

の2番目に目的の単語「研究」があがる。これを選ぶためにTouchMeKey4の左方の↓キーで「研究」をハイライトし(図1の2行目中),選択キーで選択する(図1の2行目右)。すると,候補ボックス,入力ボックスの中身が消え「研究」がテキストボックスに現れる。次に「する」についても同様に入力する。まず,2番キーで「す」を入力した後(図1の3行目左),4番キーで「る」を入れる(図1の3行目中)。ここで「テーマ」が最上位に挙がっているが,これは,さきほど入力した「研究」が,ユーザ文書中では「研究テーマ」として多用されていたためこのような結果となる。「する」を↓キーでハイライトして,選択キーで選択する(図1の3行目右)。最後に,記号「。」を4番キーに用いて同様に入力する。

本入力方式では辞書に登録されていない単語(未知語)の入力は弱点となる。この弱点は現在普及しているかな漢字変換同様のものであるため,TouchMeKey4でも未知語はかな漢字変換と同様に工夫をして入力を行わざるを得ない。たとえば,未知語の文字を含む単語を入力し,そのうえで前後の不要文字を消去することを繰り返して未知語を入力するか,あるいは1文字ずつ入力するなど作業を要する。そのうえでその単語を辞書に登録することにより,次回からの入力を短時間で行う。たとえば「情処」を入力するには「情報」と入力して「報」を消去し「処理」を入力して「理」を消し,そのうえで「情処」を辞書にユーザが登録するのである。

TouchMeKey4を実現するには,

- ユーザインタフェースの設計
- 膨大な候補の整列の方法
- かなのキーへの割当ての方法

の3つの問題を解決する必要がある。以下では順にこれらについて述べていく。

### 3. ユーザインタフェース

#### 3.1 かなを入力する方式

4つのキーでの日本語入力にもさまざまな方式が考えられる。第1に,通常のかな漢字変換の方法を踏襲して,かなを入力した後,漢字へ変換する方法である。少数のキーでかなを入力する方法にも大きく分けて2つ存在する。第1の方法は,1つのキーに割り当てられているかなの間に全順序を付け, $i$ 番目のかなを入力する際に,そのキーを $i$ 回押す方法である。たとえば,現在最も普及している携帯電話上の日本語入力方法「かな方式」がこれに相当する。たとえば「へ」を入力する場合「へ」が「は」段の4番目であるので

本稿における単語とは,日本語の形態素解析ソフトウェア「茶筌」<sup>14)</sup>による形態素解析の結果の1単位をいう。その理由については4章を参照。

ここでは,候補は5つずつ表示しているが,この数は簡単により少数にすることができ,ディスプレイの大きさに合わせる事が可能である。

6を4回押すのである。この方式の難点は、重複して割り当てられているかなが多くなると、そのかなを得るために同じキーを多数回押さなければならない点である。たとえば、2章のデバイスの場合は「へ」の入力では「な」行の5つのかなが「は」行の前にあるので、9回も3番キーを押すことになってしまう。「べ」では、14回も押さねばならず、キー数が少数であることを想定している場合、この方式は非現実的である。

第2の方法は、 $i$ 番目のかなを入力する場合には、そのキーの後に $i$ キーを合わせて押す手法である。これは「ポケベル方式」と言われ、たとえば携帯電話において「へ」を入力する際には列の6の後に「え」段の4と押すことに相当する。しかし、本稿では1つのキーに割り当てられているかなの総数は10を超えており、キーの総数よりも多いため、 $i$ に相当する数を入力することが難しい。この考え方をより一般化すると、複数のキーを同時に押すコード(chord)方式で実現することが考えられる。しかし、この方式は、速記者の例に見られるように、訓練が必要であるため、普及しにくいという問題点がある。

### 3.2 かなを入力しない方式

少数のキーを用いて入力することを前提とした場合、かなを入力する方式は、このようにさまざまな限界がある。そこで、ここでは、かなによる入力以外の方法を考えることとする。直接漢字を入力する方法には部首による入力など考えられる。しかし、現状の日本語においてはかなを用いた入力が普及しているうえ、平仮名やカタカナの入力がこの方式では問題となる。となると、かなを入力しない方式を採用するとはいえ、現行のかなを入力する方式に類似度が高いものが、受け入れられやすいと考えられ、2章で示した1かな1押し方式が候補にあがる。別の候補として、4つのキーにアルファベットを割り当てて、ローマ字入力を1かな1押しで実装するという入力方法も考えられる。しかし、街中での日本語入力では、携帯電話においてもローマ字よりもかな直々による入力は広く親しまれている。このことから、4つのキーでも比較的受け入れられやすい入力方法として「1かな1押し方式」を採用した。

「1かな1押し方式」に類する入力手法は、携帯電話上の文書入力の分野でも最近話題になっている。たとえば Tegic 社<sup>8)</sup>の T9 はヨーロッパ諸語の携帯電話での文入力システムを「1アルファベット1押し方式」で行うことを実現している。また、日本語に関しては、我々は別に研究を行っており<sup>15)</sup>、キーの数が10である場合には現実的な入力効率が得られることを示

した。本稿の研究は、この携帯電話における考え方を一般化し、さらにキー数を減らす試みであると位置づけられる。

以上の議論から、以下では「1かな1押し方式」の前提のもとで考えられるさまざまな変種について考えていく。第1は、2章で示した、キー列から結果の日本語文字列を直接推定する方式のほか、まずキー列からかな列を推定してユーザがかな列をいったん確定し、そのかな列に対してさらにはかな漢字変換を行う方式も考えられる。ユーザが2度選択する後者の方式は、携帯電話における日本語入力方法として Tegic<sup>8)</sup>や ZI<sup>10)</sup>らが提唱している。しかし、過去の検証<sup>15)</sup>によると、2段階方式と1段階方式のキー入力数の効率は、キー数が10の場合にはほぼ同じであることが分かっている<sup>15)</sup>。それにもかかわらず、ユーザが2度選択しなければならないのは、使い勝手の面で問題があると考えられる。そこで我々は、直接日本語文書を推定する前者を採用するものとした。

第2に、変換の際に補完を行うかどうかである。補完とは、ユーザの入力列に相当する接頭部分を持つ単語をすべて候補とする方法である。たとえば、2章の例では「141」というかな3文字相当分だけの入力から、「研究」というかな5文字の単語を候補とする。当然、補完を行うほうが行わない場合よりも探索空間は大きくなる。ここで、近年の小型入力機器では小さなボタンでの入力の観点からなるべく補完を採用する方向へ進んでいる<sup>4)</sup>。また、補完は中国語の入力において一般的に用いられてきており、文書入力において効果を発揮することがうかがえる。この点から、本稿では補完を採用するものとした。

最後に、入力する言語単位についても単語、文節、連文節などさまざまなものが考えられる。本稿ではすでに補完と1かな1押し方式を採用しており、ただでさえ候補の探索空間が大きい。そこで、まずは単語入力をういてどこまで入力を効率化できるかを検証することとした。

以上、4つのキーでの入力方法を議論してきたが、結果として得た方法は、1回の入力につき膨大な数の候補を扱う方法となっている。候補を得た後にその妥当性を検討し、整列してユーザに提示する必要があるので、次節では、この要となる候補の妥当性の評価方法について検討する。

## 4. 候補の整列

過去の研究<sup>15)</sup>では、キー数が10の場合に「1かな1押し方式」を採用すると、毎日新聞社会面1年分の

語彙を用いた場合に、単語相当の入力数字列に対して 32 単語弱の候補が平均してあがってしまうことを報告している。これに対して、かな漢字変換の場合には、単語相当のかな列に対して 4 単語強程度しか候補にはあがらない。一方、本稿の想定では 1 かな 1 押し方式でキー数はわずか 4 なのだから、候補数は膨大であることは間違いない。

膨大な候補を扱う場合に効果を発揮する 1 つの方法は、ユーザの入力はしよせん文脈が限定されている点に注目することである。たとえば、ゲームの入力として本入力方式を使うのであれば、文脈はそのゲームに限定されるであろうし、携帯電話であれば、ユーザ固有のメールなどの内容に限定される。とすると、語彙表現や文脈の個別化 (パーソナライズ) を行うことにより、候補選択の効率をあげることができるであろう。

ここで問題となるのは、ユーザのコーパスは、新聞コーパスなどと比較すると量をはるかに小さいということである。とすると、静的な  $n$ -gram モデルを用いて個別化を行うと、有意な統計をとれないおそれがある。そこで、まずは新聞など大容量の言語コーパスから基本となる言語の統計モデルを作り、それを少量のユーザ文書を用いて個別化することが効果的であろう。この考えに適合する枠組として PPM<sup>1)</sup> を用いた言語モデルがあげられる。

PPM はもともと算術符号化における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。このため、文脈が反映される補正の方法が十分に研究されている。さらに、わずかに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、小さなユーザ文書しか得られなくとも効果が期待できることから、本稿の問題解決には向いている。実際、かな漢字変換に対する学習方式としてはさまざまなものが考えられるが、我々の実験によると PPM によるものが一番学習効率が高かった<sup>16)</sup>。また、PPM は本入力システム以前にも、Ward<sup>9)</sup> が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。

以下、PPM を Witten<sup>1)</sup> の文献に基づき概説する。ある時系列  $S = x_0 \dots x_i$  が与えられたとき、次の要素  $x_{i+1}$  の確率を次式で推定する。

$$P(x_{i+1}) = \sum_{k=-1}^{kmax} u_k P_k(x_{i+1}) \quad (1)$$

ただし、 $k$  は  $x_{i+1}$  の前  $k$  単語のことを表し、特に  $k = -1$  のときの  $P_{-1}(x_{i+1})$  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する初期確率を表現する。 $k \geq 0$  の場合の  $P_k(x_{i+1})$  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、 $c_k(x_{i+1})$  をその文脈で  $x_{i+1}$  が現れた回数とすれば、

$$P_k(x_{i+1}) = \frac{c_k(x_{i+1})}{C_k} \quad (2)$$

と近似される。

$u_k$  は直前の  $k$  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<sup>1)</sup>。本稿の問題の場合には、その差を調査したものの、有意な差は得られなかったため<sup>16)</sup> PPM を用いるものとした。具体的には、 $e_k$  を

$$e_k = \frac{1}{C_k} \quad (3)$$

また、 $l$  ( $\leq kmax$ ) をコーパス中の単語列で、現状の文脈に適合する最長の単語列の長さとして、

$$u_k = (1 - e_k) \times \prod_{i=k+1}^l e_i \quad -1 \leq k < l \quad (4)$$

$$u_l = 1 - e_l \quad (5)$$

により定義する。 $k$  の上限  $kmax$  は 4 (5-gram) とした。

学習は、ユーザ文書の最後尾にユーザが入力する単語を次々に付加し、ユーザ文書の大きさを徐々に大きくすることにより行う。ユーザ文書の大きさはこのように時々刻々と変化するため、確率  $P(x_i)$  の推定は使われるたびに再計算する。この計算を動的に行うために、ユーザ文書は Suffix Array<sup>3)</sup> のデータ構造により保持している。その分空間量が必要となるが、そもそもユーザ文書は大きくても数万単語で十分であるため (6 章参照)、基本辞書に比べると数分の 1 で済む。

本稿では  $k = -1$  の場合の単語辞書は、毎日新聞の社会面から構築する。社会面は、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ため、このような辞書を構築するのに適している。コーパスは形態素解析し、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

が記述されている。本稿で入力の単語の単位が「茶筌」<sup>14)</sup> の解析結果の単位であるのは、この工程のため

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。

表 1 キー割当て  
Table 1 Various key assignments.

キー	割当て	条件付き エントロピー	「にほんご」 を入力
11-key-a	各段に 0~9 濁音, 半濁音, 記号は*	1.807	2602*
10-key-a	各段	2.373	2602
5-key-a	あは段 (1) かま段 (2) さや段 (3) たら段 (4) なわ段 (5)	3.173	5152
5-key-b	あま段 (1) から段 (2) さな段 (3) たは段 (4) やわ段 (5)	3.453	3452
5-key-c	あか段 (1) さた段 (2) なは段 (3) まや段 (4) らわ段 (5)	3.737	3351
5-key-v	あ行 (1) い行 (2) う行 (3) え行 (4) お行 (5)	3.599	2555
4-key-a	あな段 (1) かまや段 (2) さわ段 (3) たはら段 (4)	3.817	1432
4-key-b	あな段 (1) かは段 (2) さまや段 (3) たらわ段 (4)	4.012	1242
4-key-c	あか段 (1) さた段 (2) なはま段 (3) やらわ段 (4)	4.051	3341
3-key-a	あたま段 (1) かなや段 (2) さはらわ段 (3)	4.600	2332
3-key-b	あなや段 (2) かさは段 (2) たまらわ段 (3)	4.763	2232
3-key-c	あかさ段 (1) たなは段 (2) まやらわ段 (3)	4.821	2231
2-key	あ~な段 (1) はやまらわ段 (2)	5.994	1221

である。基本辞書には、約 8 万単語の異なる単語が含まれる。

また、 $k = 0$  以上の場合は、ユーザ文書から統計を動的に作成する。ユーザ文書については本稿では 2 つの文書に関する結果を示す。第三著者の本の原稿（単語数 7 万）と、1 年分の電子メール（総単語数 16 万）のである。これらは、一部を切り出して学習とテスト文書として使用した。ユーザ文書は局所的な偏りをなくするために、文章ごとに切り分け、これをランダムに並べ替える。実際の入力時は、入力したい文章は局所的な偏りがあるものであるが、本稿では 6 章においてある一定の学習量における入力効率を均質的に計測するためにこのようにした。

## 5. かなのキーへの重複割当て

キーの総数が少ない場合、どのようにかなをキーに重複して割り当てるかによって、文書入力の効率は異なる。たとえば、4 キーの場合「あ」段から「は」段までを 1 つのキーに割り当て、残りの「や」「ら」「わ」段に割り当てる場合と、「あ」「は」段、「か」「ま」段、「さ」「や」段、「た」「ら」「わ」段をそれぞれ 1 つずつのキーに割り当てる場合では、後者の割当てのほうが単語推定が容易のように思える。なぜなら、前者の割当てではキーへの割当ての数に明らかな偏りがあるからである。とはいえ、後者の方式と「あ」「か」段、「た」「さ」段、「な」「は」「ま」段、「や」「ら」「わ」段をそれぞれ 1 つずつのキーに割り当てる方式のどちらがより適切かは、自明ではない。

むろん、6 章で示すように、キー入力が最終的に最少となるものを実験的に選択することも可能である。しかし、キー入力実験はテスト文書の量や種類に結果が依存してしまう。何よりも、デバイスの設計はあま

り用途に特化したものに限定してしまうよりも、なるべく汎用なものにし、あとは PPM による学習で調整するほうがよいであろう。

ある単語相当の数字列が与えられたときに、単語を平均的に適切に推定することは、 $W$  を単語、 $C$  を入力文字列として、条件付き確率  $P(W|C)$  がよりの確に推定できることに相当する。そのときには、条件付きエントロピー、

$$H(W|C) = -\sum_{w,c} P(C=c, W=w) \log P(W=w|C=c) \quad (6)$$

が、より小さくなると考えられる。すなわち、上式の値が小さい場合のほうが、 $C$  が与えられたときの  $W$  がより不確か (less uncertain) でなく推定できるということである。

つまり、上式は評価関数の 1 つの候補となりえる。そこで、言語モデルを特定し、上式が最小となる割当てを探索することにより、その言語モデル下での最適な割当てを探索することが考えられる。ここで、割当ての総数は、かな総数のキー総数乗だけ考えられるため、最適化はしよせん近似となってしまう。また、日本語の「かな」には慣例的な体系があるから、これに準じた割当てを行うほうが、ユーザの使い勝手の観点からはよいであろう。つまり、日本語の場合には 50 音表の体系はユーザの観点からは重要であり、子音段や母音行の概念を単位とした割当てにしたほうがよい。たとえば、段を越えての割当て、たとえば「あ」「ち」「け」「ゆ」「ま」「は」「よ」を第 1 のキーと割り当てるのでは、いかにも使いにくいそうである。

そこで、段や行を単位としたさまざまな割当てを考え、単語頻度だけを考慮した言語モデル下での条件付きエントロピーを表 1 に示す。この表の第 1 列はキー

割当ての名前，第 2 列はキー割当て，第 3 列は割当てのエントロピー，第 4 列は日本語を入力する場合のキー割当てを示している．エントロピーは，毎日新聞（94 年）の社会面に対して算出した．濁音や半濁音は 11key 以外は濁音のない場合のかなと同じキーに重複させて割り当て，また記号と「ん」はわ段に含めている．

まず，キーの数が減るにつれ，エントロピーが増加していることが分かる．また，同じキーの数の間でも割当てによってはエントロピーが大きく変動する．いずれの場合も，割当て a が表の中ではより良い割当てであることがわかる．なお，図 1 は 4-key-c の割当てを用いた場合の入力を採用した場合の例である．かなの体系からは前から順に割り当てただけであるので，非常に分かりやすいが，エントロピーの観点からはさほどよくはない割当てであることが観察される．

## 6. 評価

### 6.1 キー入力数による評価

本節では評価としてまずキー入力数を計測する．これは，4 章で示した PPM による言語モデルの性能，すなわち TouchMeKey4 システムの候補推定のよさを計測することに相当する．特にここでは補完を用いているため，キー入力数がそのまま候補予測の性能に直結する．

ある量の文書を学習した際に，同種のテスト文書を入力するのに必要な単語あたりのキー入力数を自動入力により計測した．学習のためのコーパスは，学習量に応じた入力効率を調べるため，小容量のものから大容量のものを用意した．

テスト文書には，学習に用いていないユーザ文書の一部を用いる．テスト文書は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 1 単語あたりの平均キー入力回数を計測した．このようなテスト文書は原稿については 1,363 単語，メールに関しては 1,587 単語を使用した．

キー入力数は，入力，候補選択に要したキー操作回数の総和である．入力と候補選択のキー入力に必要な時間は必ずしも同じではないが，前述のようにこの節は入力の時間に焦点を当てているのではなく，システムの候補推定能力を測る目的であるため，この値を

単語の長さに相当する分を入力しきってから候補を表示して選ぶ場合には，候補が何番目にあがるかだけを見ればよいが，補完の場合には，単語の入力を進めるにつれて候補の順位が変わるため，キー入力数そのものが言語モデルのよさの 1 つの尺度となる．

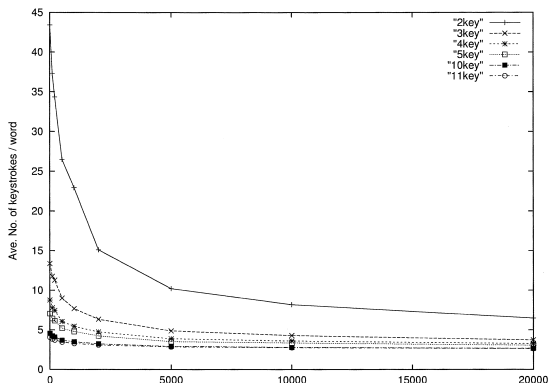


図 2 学習量と単語あたりの打鍵数（原稿）

Fig. 2 Average number of keystrokes/word (textbook).

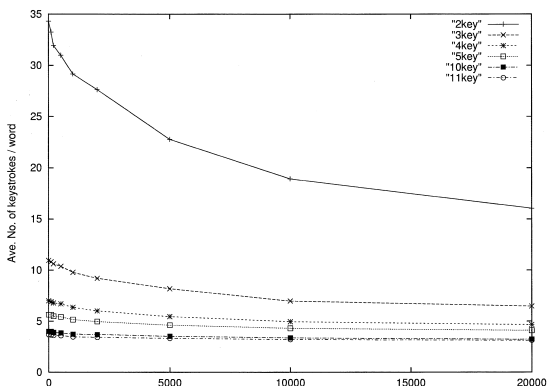


図 3 学習量と単語あたりの打鍵数（メール）

Fig. 3 Average number of keystrokes/word (e-mail).

用いることができる．選択については  $n$  番目にあがった候補を選択するためのキー入力数は  $n$  回であるものとした．たとえば，図 1 の「研究」の入力にかかったキー入力数は，入力が 3，選択が 2 で計 5 である．なお，補完を用いた単語変換では，入力の仕方は一意ではない．たとえば「研究」についてももう 1 文字「ゆ」相当の「4」を入力すると，候補は第 1 候補にあがる．そこで，自動入力に際しては，次のようにする．かな相当の入力の途中で目的の候補が第 1 候補にあがった場合はその時点で単語を選択し，第 1 候補にあがらなければ，かな相当の入力を続行する．つまり，図 1 の「研究」のように候補があがった時点で選択することはせず「ゆ」の入力を行って，第 1 候補としてあがって初めて選択される．

図 2，図 3 にユーザ文書の量と単語あたりの入力数との関係を示す．ユーザ文書は，前者が原稿，後者がメールである．横軸がユーザ文書の単語数，縦軸が単語あたり平均打鍵数であり，さらに線は下から順に，

表 2 各キー数における入力効率

Table 2 Text entry efficiency according to the different number of keys.

キー数	原稿 (5万単語)	メール (15万単語)
11	2.62	2.88
10	2.63	2.91
5	2.92	3.55
4	3.10	3.68
3	3.43	4.65
2	5.58	10.13

キー数が 11, 10, 5, 4, 3, 2 の (表 1 の各キー数における a の割当てを用いた) 場合である。

まず, 学習がない場合には, 多くの打鍵が必要となる。一方, 2 万単語の学習を行うと, メールの場合にはキー数が 4 ならば 5.0 を下回り, また原稿の方はキー数が 3 でも 5.0 よりも小さい。一般に PC 上のキーボードを用いた日本語入力, ローマ字入力 (つまり, 1 かな 2 押しの, 補完を用いない現行の入力方式) が主で, 連文節変換により入力されるが, この方式を用いてテスト文書の入力を行ってみると, 原稿のときには 1 単語あたり 5.75, メールときには, 5.77 のキー入力数が必要であった。したがって, 単語あたり平均打鍵数が 5.0 を下回るといことは, PC 上の現行のキーボード入力の打鍵数の程度には PPM による言語モデルが効果を発揮していることになる。

表 2 にユーザ文書を限界まで増やしたときに, 各単語あたりの平均キー入力数を示す。ユーザ文書は, 原稿は 1 冊の本原稿の 5 万単語, メールは 1 年分 (15 万単語) を用いた。これを見ると, メールも原稿も, キー総数 3 でも 5.0 を下回る。

以上から, 学習を用いると, 本稿の範囲では 3 や 4 といった少数のキー数でもでも PPM を用いれば, 現行のシステムと同程度の打鍵数で, 候補推定を行うことができることが実験的に示された。

なお, 本稿では 2 つのユーザ文書についてのみ結果を示したが, このほか, 新聞の経済面を用いた学習実験でも同じ結果が得られている。また, 英語についても, 3 種のユーザ文書を用いた実験を行ったところ, 4 つのキーでは現実的な候補推定を行うことができることがやはり示され, その際のユーザ文書の学習量も 1

単語の平均長はメールの場合には 2.33, 原稿の場合には 2.25 であった。これよりも, 単語あたりの打鍵数が少なくなるのは, 補完を用いているからである。

本稿がキー総数 4 の場合の例を多用するのは, 1 万単語の学習を行えば, どの言語でも, どのコーパスでも 1 単語あたりの打鍵数がフルキーボードの打鍵数を下回ったからにすぎない。

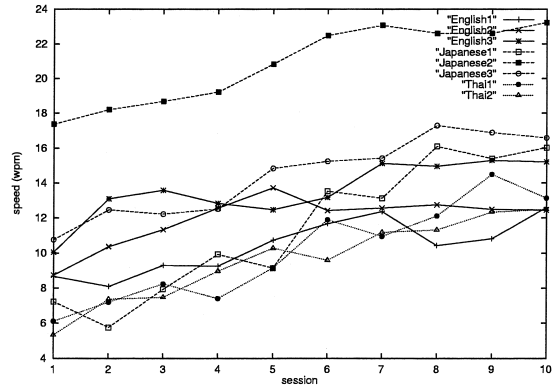


図 4 実際のユーザによる入力速度

Fig. 4 Speed.

万単語程度と同等の結果が得られた<sup>7)</sup>。

## 6.2 ユーザ実験

前節の評価は言語モデルの評価であり, ユーザの視点からの評価ではない。そこで, 本節ではユーザに実際に使ってもらうことにより, 使い勝手を評価した。現在, TouchMeKey4 は日本語のほか, タイ語, 英語での入力を行うことができる。タイ語については, 4 つの各ボタンに声調を割り当てて入力を行う。また, 英語の場合には 4 つのキーのそれぞれに ABCDEF, GHIJKL, MNOPQRS, TUVWXYZ と記号, と割り当てている。

本稿では日本語に焦点を絞ってここまで報告を行ってきたが, ユーザ入力については 3 言語合わせて報告する。8 人のユーザに TouchMeKey4 での入力を依頼した。3 人が日本語, 3 人が英語, 2 人がタイ語を担当した。このうち, 英語を入力した 2 人以外は全員母国語での入力を行った。

各ユーザは TouchMeKey4 の GUI に対応する 4 つボタンが左右についたゲーム用のコントローラを用いて, 与えられた文章を入力する。文章はユーザ文書に存在する単語のみからなる, ユーザ文書中にない文章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のみ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 4 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示した。速度は, 1 分あたりの入力単語数 (wpm) で計測した。各線は被験者 1 人に相当し, 実線が英語, 細点線がタイ語,

英語を母国語とする被験者が一番入力速度が遅かった。



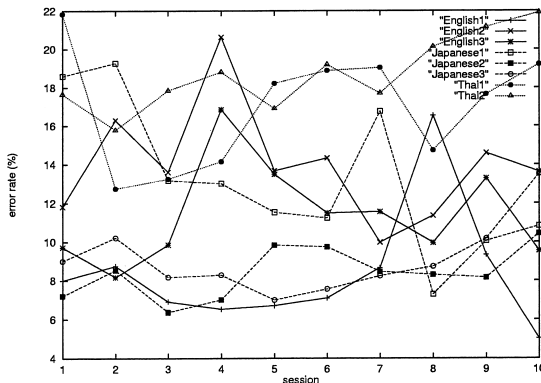


図5 エラー率  
Fig. 5 Error rate.

長点線が日本語による実験を表す。

すべての被験者が5時間後には12wpmを超えた速度で入力ができるようになった。また、図によれば、1人だけ入力が特に速い被験者がいる。この被験者は実験に入ったときにはすでにTouchMeKey4での経験を積んでいたため、このような記録を達成した。30分間の最高平均入力速度は23wpmであり、1単語の長さは約2.2字のメールの場合では、10分間に約500字入力できることとなり、ワープロ検定では2級程度である。

また、グラフのすべての折れ線が右上がりになっていることから、全員が5時間のトレーニングにより入力速度を上げている。当初は6~8wpmであった初心者も5時間後には16wpmを越える速度で入力を行っており、TouchMeKey4ではユーザによる学習が入力速度を向上させることができる。

言語別の比較を行うと、日本語が最も速い。条件付きエントロピーでは、英語は1.3、タイ語は1.1であるので、日本語が最も入力上の複雑度は大きい。にもかかわらず、日本語の入力速度が速いのは、日本人が日頃かな漢字変換を使い慣れているためであろう。

図5には、同じ被験者たちの平均エラー率を示した。ここでも各線は被験者1人に相当し、実線が英語、細点線がタイ語、長点線が日本語による実験を表す。横軸はセッションで、縦軸はエラー率である。エラー率は、一度でも取消キーを押した単語は実際にエラーでなくともエラー単語と見なし、エラー単語の全単語の比率とした。このグラフからはエラー率は10~20%の範囲である。入りに慣れたからといってエラーが減るとは限らないようで、人によっては入りに慣れ

ると、却って選択時のミスが増える。また、エラー発生率は個人差も大きい。大量の間違いをしつつ入力を行う被験者もいれば、ほとんど間違いをしない慎重派の被験者もみられた。

10~25wpmというのは確かに遅く、エラー率も人によっては大きい。とはいえ、同じ(未知語を除いた)テスト文書に対して、10個のキーを補完を用いて入力すると、

- 「かな方式」の場合には同程度、
- 「1かな1押し方式」での入力速度は20~33wpm、との実験結果を文献15)で報告している。特に前者の方式は現行の携帯電話での入力システムとして採用されている方法でもあることから、現行の携帯電話上のシステムに匹敵する入力速度が得られているといえることができるであろう。

## 7. 結 論

本稿では4つのキーを用いて自由な日本語の文章を入力するシステムを設計した。さまざまな実験の結果、4つのキーを用いたとしても、学習を用いれば現行の携帯電話における文書入力に匹敵する効率で入力が可能であることが分かった。

入力方式としては、かなを重複して1つのキーに割り当て、そのキーを1つのかなにつき1回だけ押す。この入力を単語単位で行い、さらに補完を用いてユーザは文章を入力する。単語の推定には動的な統計言語モデルを用いた。さらにユーザ文書の学習を行うことにより、想定される入力文書に単語推定を特化する。

本稿の応用としては、小型機器類への搭載のほか、タッチパッドや視線検知システムなどと組み合わせ、高齢者や身体障害者に優しい文入力システムが考えられる。今後はこれら応用システムに挑戦し、本システムの有効性をさらに示していきたい。

## 参 考 文 献

- 1) Bell, T.C., Cleary, J.G. and Witten, I.H.: *Text Compression*, Prentice Hall (1990).
- 2) James, C. and Reischel, K.: Text input for mobile devices: Comparing model prediction to actual performance, *Proc. Annual SIGCHI Conference*, pp.367-371 (2001).
- 3) Manber, U. and Myers, G.: Suffix arrays: a new method for on-line string searches, *Society of Industrial and Applied Mathematics Journal of Computing*, pp.319-327 (1993).
- 4) Masui, T.: PObox an efficient text input method for handheld and ubiquitous computers, *Proc. International Symposium on Hand-*

*held and Ubiquitous Computing*, pp.289–300 (1999).

- 5) Mathias, E., MacKenzie, I.S. and Buxton, W.: One-handed touch typing on a qwerty keyboard, *Human-Computer Interaction*, Vol.11, No.1, pp.1–27 (1996).
- 6) Noyes, J.: Chord keyboards, *Applied Ergonomics*, pp.55–59 (1983).
- 7) Tanaka-Ishii, K., Inutsuka, Y. and Takeichi, M.: Entering text with a four-button device, *International Conference on Computational Linguistics*, pp.988–994 (2002).
- 8) Tegic, 2000. <http://www.t9.com>
- 9) 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*, pp.129–137 (2000).
- 10) ZI, 2000. <http://207.229.18.241/>
- 11) ミサワホーム . Cutkey, 2002. <http://www.misawa.co.jp/CUTKEY/index.html>
- 12) 杉本正勝：片手操作キーカード (shk) による日本語入力, 情報処理学会モバイルコンピューティング研究会論文集, Vol.1, pp.1–8 (1997).
- 13) 増井俊之：インターフェイスの街角 (14)—ウェアラブルコンピュータのテキスト入力方法, *Unix Magazine*, Vol.14, No.2, pp.2–9 (1999).
- 14) 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).
- 15) 田中久美子, 犬塚祐介, 武市正人：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, 情報処理学会論文誌, Vol.43, No.10, pp.3087–3096 (2002).
- 16) 武市正人, 田中久美子, 丸山卓久：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, pp.9–14 (2001).

(平成 13 年 10 月 25 日受付)

(平成 14 年 12 月 3 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の研究に興味を持つ。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。