

重度身障者のための1ボタン自然言語入力システム

田中久美子

東京大学

113-8658 文京区本郷 7-3-1

kumiko@r.dl.itc.u-tokyo.ac.jp

Abstract

本稿では、モールス信号に基づく重度身障者のための1ボタン入力システムを報告する。単なるモールスコードに基づく入力システムに、単語の予測を導入し、さらに予測された単語候補のうち、候補選択により入力時間を削減しないものの刈り込みを行なったところ、入力効率の大きな向上が見られた。本工夫は、現在では英語での効果のみ検証されているが、日本語のシステムや、一般の予測入力システムにも適用可能である。

1 はじめに

計算機技術は、健常者だけでなく、身障者の環境をも大きく変化させている。一般に、身障者の環境を改善する問題は、健常者用のそれよりも大きな制約があるため、難しいことが多い。しかし、それを解くことにより、身障者だけでなく、健常者の問題に対して新たな提言を与えることがある。本稿では、1ボタン入力という極度に制限された状況下での入力を考えることにより得た、一つのユニバーサルな技術となりうる言語処理例を示す。

本稿の問題は、1つのボタンだけを用いた入力である。このようなシステムは、身障者の中でもALS患者や、頸椎損傷を負った患者により実際に使われている。これまでも、国立身体障害者リハビリテーションセンターでの研究 (Itoh, 1997) や、台湾 (Wu and Luo, 2002)、他に米国での研究も見られる。既存の研究では、モールスコードを用いたものがほとんどであるが、未だ予測入力は導入されておらず、ユーザは長い単語であっても最後までモールスコードを用いて入力しきらなければいけないという問題があった。そこで、本稿では既存の研究に、

- 予測入力機能を付け加え、
- 候補選択により入力時間が短縮されないものを刈り込む

表 1: モールスコードにおける入力の意味

時間	入力	間
1 ユニット	•	入力間
3 ユニット	—	文字間
7 ユニット	未定義	単語間

工夫を行なった。実際に英語に関してこのようなシステムを実装し、どの程度の入力効率の向上が見られるかを検証した。

2 システム概要

1ボタンで入力を行なうには、まずは1ボタンの出力を、言語のアルファベットなどに何らかの方法で展開する必要がある。その方法はさまざまに提案されているが、本稿では過去の多く試みと同様、モールスコードを用いるものとした。モールスコードはアルファベットの単純な文字頻度をもとに作られている。とはいえ、今でも無線通信において事実上の標準として使われている上、プロの入力は1分間あたり25単語と高速な入力を可能とすることから、本入力においてもこれを基本とした。

モールスコードを用いた入力は、ボタンを異なる長さで押し分けることにより行なわれる。入力は最短の長さを1ユニットとすると、3ユニット分の長さとして7ユニット分の長さが定義されており、その長さ分の入力と間(入力しないこと)それぞれに意味付けが行なわれている(表1参照)。通常アルファベットの入力には、3ユニット分の長(本稿では—として記述)と短(•)を表2の表にしたがって押し分けることにより行なわれる。「間」に対する意味付けとしては、入力間には最低1ユニット分の間を空ける必要があるし、3ユニット分の間を空けると文字間の間、7ユニット分の間を空けると、単語間の間と解釈される。たとえば、「SOS」の入力は、

•(休み1)•(休み1)•(休み3)

表 2: モールスコードによるアルファベット

char	code	char	code
A	● —	N	— ●
B	— ● ● ●	O	— — — —
C	— ● — ●	P	● — — ●
D	— ● ●	Q	— — ● —
E	●	R	● — ●
F	● ● — ●	S	● ● ●
G	— — ●	T	—
H	● ● ● ●	U	● ● —
I	● ●	V	● ● ● —
J	● — — —	W	● — —
K	— ● —	X	— ● ● —
L	● — ● ●	Y	— ● — —
M	— —	Z	— — ● ●

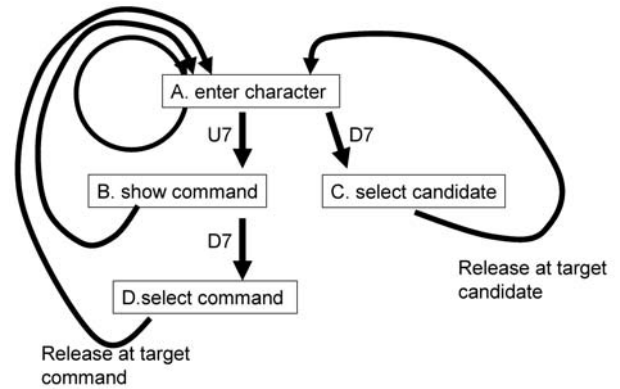


図 2: システムにおけるモード変化

ように、本システムでは、単語の prefix 部分を入力すると単語全体が候補として挙がる予測を用いて入力を行なう。

候補ボックスから望みの候補を選択するには、ユニット 7 の長さで press ボタンを長押しする。すると、候補ボックスの候補のハイライトが始まり、一定の速さで候補を走査する。ハイライトされているときに、ボタンを放すと、その候補が入力される。

このやり方は、左の画面で、コマンドを選ぶことにおいても同様である。左の図で Press ボタンを 7 ユニットの長さで押すと、候補ボックスの New Line コマンドがハイライトされる。さらに押しつづけると、つぎつぎにコマンドがハイライトされる。目的のコマンドのところ Press ボタンを放すと、そのコマンドが実行される。モードを切り替えを 7 ユニットで押しつづけることにより行なうことは、それがもともとのモールスコード上で未定義であったことを利用している (表 1 参照)。

ボタンを押すことによって、どのようにモードが変化するかを図示したのが、図 2 である。D7 は 7 ユニット分ボタンを押している状態、U7 は 7 ユニット分ボタンを放している状態を表している。一般に、モードには選択モードと、入力モードの二つがあり、通常は入力モードとなっている。選択モードに切り替えるには、Press ボタンを 7 ユニット以上押しつづける。選択モードから入力モードへは、候補ボックスから選択が行なわれると自動的に入力モードへと切り替わる。

本予測入力システムの特徴は、辞書にない単語であっても、入力が可能である点である。というのも、モールス信号で文字を入力することは、常に可能であるからである。一般に辞書にない単語をいかに入力す

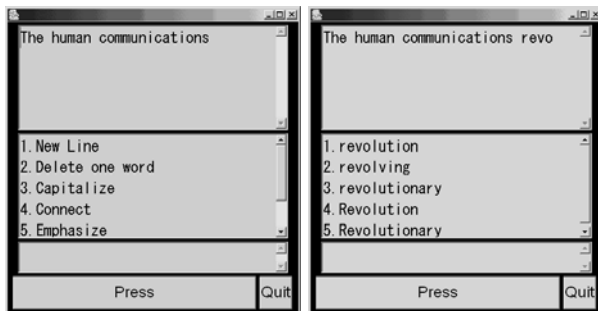


図 1: システムの GUI. コマンドモードと、補完モード

—(休み 1)—(休み 1)—(休み 3)

●(休み 1)●(休み 1)●(休み 7)

と入力する。以上はモールスコード上の定義である。

図 1 に、Java 言語で実装された本入力システムの GUI を示す。下方の Press ボタンを押すことにより入力が行なわれ、結果のテキストが上方の文書ボックスに、中央の候補ボックスにはユーザが選択可能なコマンドが、下方横長の窓 (Press ボタンのすぐ上) には、現在ユーザが入力したモールス信号が表示される。左図においては、すでに “The human communications” という文章が入力された状態を示す。通常は、コマンドモードとなっており、候補ボックスにはコマンドが表示されている。

ユーザは、ここで “revolution” の入力を行なうものとする。表 2 にしたがって、アルファベットの入力を “revo” まで行なったところの図が右図である。候補ボックスには目的の候補が 1 位に挙がっている。この

るかという点は、予測入力やかな漢字変換においては最大の論点となるため、本システムはこの点が一つの利点となっている。とはいえ、特に長い単語などは予測を用いた方が入力の効率を向上させられる可能性があるし、本稿では重度身障者を対象としているため、予測を用いることによる効果が期待される。以下では予測の効果を確かめる。

3 予測入力

3.1 基本モデル

予測では一般に入力 S に対して最大の条件付確率を与える語 W を最良の候補とする。

$$\operatorname{argmax}_W P(W|S) = \operatorname{argmax}_W P(S|W)P(W) \quad (1)$$

$P(S|W)$ は入力誤りがないと仮定すると 1.0 となるため、結局 $P(W)$ 、すなわち言語モデルが候補の良し悪しを決定する。 $P(W)$ の予測についてはさまざまなものがあり、単純な頻度から、高次の n-gram を利用するもの、動的に n-gram を推定する PPM を利用するもの、品詞列を利用した HMM を用いるものなどさまざまなものが提案されている。

本稿では、最も単純な頻度によるモデルを用いる。というのも、つぎに述べる刈り込みを論じるには、高次のモデルよりは、単純なモデルの方がそのふるまいが明確であるからである。とはいえ、次章で述べる刈り込みは、高次の言語モデルを用いたとしても併用できるものである。

3.2 候補の刈り込み

刈り込みによる工夫の基本アイデアは (Tanaka-Ishii and Frank, 2004) に述べられているものである。一般に入力においては、それぞれの候補の入力にかかる時間は異なる。たとえば、“sh” が入力済みで、she, shy, should を入力したい場合について考えてみる。she の場合には、e を入力するためには、1 ユニット (e の入力分) しかかからない。一方で、候補から選択する場合には、モードの切り替えだけで最低 7 ユニットかかってしまう。したがって、she については、ユーザが e を直接入力してしまった方が早いことになる。

一方、shy であるが、y の入力は、13 ユニットである。というのも、—•— は、長に 9 ユニット、短に 1 ユニット、そして、長と短の間に 3 ユニットかかるからである。同じように考えれば、should は、47 ユニットかかる。というのも、o は — — — は 11 ユニッ

ト、u は ••— で 7 ユニット、l は —•— で 13 ユニット、d は —••7 ユニットで、さらに文字間に 3 ユニットづつおかなければならないので、 $3 \times 3 = 9$ ユニット足して計 47 ユニットである。とすると、should は予測により入力を行なう方が直接入力を行なうよりも、時間を短縮できそうであることが予想される。

一般に、現在ユーザの入力した文字列を prefix とする単語 s を選択するのにかかる時間を、 $t_{select}(s)$ 、suffix 部分を入力する時間を $t_{enter}(s)$ とする。すると、

$$t_{enter}(s) < t_{select}(s) \quad (2)$$

が成り立つ場合には、候補リストから s を削除することにより、入力が困難な単語の候補順位が繰り上がり、さらに入力時間を短縮できる。そこで、ユーザに候補を提示するたびに、候補を高順位のものから走査して、上の条件を満たす候補を削除する。

$t_{enter}(s)$ については、should で示した例と同様に計算する。一方で、 $t_{select}(s)$ については、その候補が i 番目の候補として挙がっているものとする、

$$t_{select}(s) = 7 + (i - 1) \times dt \quad (3)$$

として計算する。ただし、7 ユニットはモード切替の時間、 dt を i 番目から $i + 1$ 番目の候補に移動するユニット数としている。

一般に、刈り込みの効果は、ユーザが見る単語の数 ($NthMax$) によって異なる。 $NthMax$ が 1 である場合には、刈り込みの効果はあまり得られない。一方で、 $NthMax$ が大きい場合には、刈り込まれて繰り上がる候補が多くなる一方で、たくさんの候補をユーザが走査することになる。この点に関する詳細は、(Tanaka-Ishii and Frank, 2004) において分析されている。

本アイデアは携帯電話における通常の英語入力や、かな文字列の日本語入力にも適用できることから、ユニバーサルな技術として発展させられる可能性があり、今後の課題となっている。

4 実験

本節では、予測入力の効果、および刈り込みの効果を計算機上の実験で確かめる。

まず、予測に用いる辞書には、WSJ の 30M バイト分の全単語を用いた。入力用のデータには、表 3 の第一列に示した 4 つのものを用意し、うち 5000 単語程度を入力した。候補をスキャンする速度 dt は 3 ユニットとして設定した。

表 3: 単なる予測入力、および刈り込みによる効率の向上 (ユーザに示す候補数が 5 のとき)

テストコーパス	入力単語数	入力時間の短縮率		キー入力数の減少率		単語の選択率	
		予測 + 刈込	予測のみ	予測 + 刈込	予測のみ	予測 + 刈込	予測のみ
ACL03 の論文	4287	18.60%	14.67%	34.8%	35.8%	59.51%	77.96%
トムソーヤ - の冒険	4714	13.60%	8.08%	28.76%	29.99%	50.36%	71.64%
アメリカ合衆国憲法	4973	17.5%	13.59%	36.80%	37.35%	67.04%	82.99%
個人の電子メール	6567	18.04%	13.84%	35.68%	35.84%	59.08%	75.22%

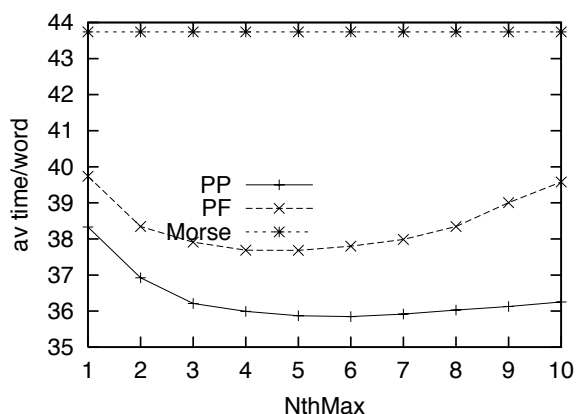


図 3: ユーザに示す候補数に対する 1 単語あたり入力に必要な時間 (電子メールの場合)

ユーザに示す候補数を 5 とした場合の結果を、表 3 に示す。まず、第四列を見ると、予測だけでも、15%程度の時間上の効率向上が見られる場合がある。最終列を見ると、上位 5 単語を見せるだけで、実に 8 割近くの単語が候補から選択されていることがわかる。その前の列 (第 7 列) を見ると、刈り込みを行なうと単語の選択率が減ることがわかる。刈り込まれた単語は候補には挙がらないため、当然の結果である。しかし、第三列と第四列を比較すると、入力時間は 5%前後早くなっている上、第五、六列の比較から、この効果が、キー入力数をほとんど変更せずに得られていることがわかる。

つぎに、ユーザに示す候補数を変更した場合の時間効率を図 3 に示す。横軸は、ユーザに示す候補数、縦軸は 1 単語あたりの入力に必要なユニット数である。折れ線は、上からモールス信号のみの場合、予測のみの場合、予測と刈り込みを行なった場合である。予測のみでは、ユーザに示す候補数が増やしすぎると、時間効率が悪くなることとなることがわかる。一方で、刈り込みを用いると、時間効率が向上しない候補が削除されるので、示す候補を多くしても時間効率が悪くはならないことがわかる。

5 結論

本稿では、モールス信号に基づく重度身障者のための 1 ボタン入力システムを報告した。既存のモールス信号に基づく入力システムの研究に加えて、本稿では新たに予測入力を導入し、さらに候補のうち入力時間の短縮をもたらさない候補を削除する刈り込みを行なったところ、入力効率の向上が見られた。本工夫は、現在のところは英語での効果のみ検証されているが、日本語のシステムや、一般の予測入力のシステムに適用可能であるため、この点が今後の課題となる。

本稿の研究へのそもそもの動機は、2003 年度にフランスで論争となった安楽死問題に端を発する。(Humbert, 2003) の著者は、交通事故になって全身不随となった後も、親指だけを用いて安楽死の権利を訴えた。いかに過酷な状況になろうとも、人間同士のコミュニケーションが重大であることを本参考文献を通して痛感したことが、本研究に結びついた。

References

- V. Humbert. 2003. *Je vous demande le droit de mourir*. Michel Lafon. ISBN: 2840989921.
- K. Itoh. 1997. Evaluation of display-based morse keyboard emulator. In *Proceedings of 12th Japanese Conference of Advancement of Rehabilitation Technology*, volume 12, pages 113–116, August.
- K. Tanaka-Ishii and I. Frank. 2004. Dit4dah: Predictive pruning for morse code text entry: towards an entry system for the seriously impaired. In *International Joint Conference on Natural Language Processing*, page to appear.
- C.M. Wu and C. H. Luo. 2002. Morse code recognition system with fuzzy algorithm for disabled persons. *Journal of Medical Engineering & Technology*, 26(5):202–207.