

行列スキャン入力方式に組み込む予測文字の配置の検討

佐々木 大真, 小山 智史

弘前大学 教育学部

〒 036-8560 弘前市文京町 1 番地

E-mail: koyama@cc.hirosaki-u.ac.jp

あらまし 重度肢体不自由者用の文字入力方式として行列スキャン入力方式があるが、効率は悪い。現在の文脈から次文字を予測し、文字盤の上部に置く方法は改善策のひとつである。ここでは、PPM法で得られた確率から、次文字を選択する際の操作回数の期待値を求め、これを小さくする予測文字を置く方法を示す。また、予測文字の置き方が入力効率に及ぼす影響をシミュレーションにより調べた結果、行配置に改良を加えた方法(行配置⁺)が広い範囲で良好な性能を示すことがわかった。

キーワード コミュニケーションエイド, 入力装置, スキャン, 高速化, 予測

A discussion about the layout of predicted letters built in a row-column scanning array

Daiji Sasaki, Satoshi Koyama

Faculty of Education, Hirosaki University

1 Bunkyo cho, Hirosaki-shi, 036-8560 Japan

E-mail: koyama@cc.hirosaki-u.ac.jp

Abstract Row-column scanning method, which is used by the severely motor disabled, is desired to reduce switch activations for input letters. In this paper, we pose the method to determine the letter to be placed in a particular cell of the scanning array automatically, in order to lower the expected value of activations to select a next letter. Then this mechanism is evaluated focusing on the layout of those predicted letters. We find that the layout which places predicted letters as a row with modification performs well in wide range of the number of input letters.

Keywords Communication aid, Input device, Scanning, Acceleration, Prediction

1 はじめに

重度肢体不自由者用の入力方式として行列スキャン入力方式がある。これは1~2個のスイッチを用いて文字盤から文字選択をするものであるが、選択に多くのスイッチ操作(または多くの時間)を必要とし、効率の改善は重要な課題のひとつである。

情報源符号化法のひとつである PPM 法は、入力テキストの文脈情報を収集しながら、現在の文脈に基づき次文字の確率を求め、算術符号化することによって優れたテキスト圧縮を実現している [1, 2]。この符号化法は比較的軽度の肢体不自由者向けの入力方式に応用され、Darragh らにより Reactive Keyboard が開発された [3, 4]。文献 [5] では、PPM 法を用いて予測した文字を行列文字盤の先頭行に並べて予測行とすることにより、入力効率が改善されることを示した。

ここでは、PPM 法で得られた確率から、次文字を選択する際の操作回数の期待値を求め、これを小さくする予測文字を置く方法を示す。また、予測文字の置き方が入力効率に及ぼす影響をシミュレーションにより調べた。なお、入力効率は、2個のスイッチを使用した場合(手動スキャン)の1文字当たりのスイッチの平均操作回数で評価した。

2 文字の予測と置き方

2.1 予測文字の置き方

行列スキャン入力方式では、文字盤左上に置かれた文字ほど、選択に必要な操作回数は少ない(図 1(a))。従って、出現確率の高い文字を文字盤の左上に置くことが有効である。

広く用いられているアルファベット順配列の文字盤は、文字の出現確率について考慮されていないため、改善の余地がある。

改善策のひとつに、現在の文脈から次文字を予測し、「有力な候補文字」数文字程度を文字盤左上に置く方法がある。以後、これらを予測文字と呼ぶ。1文字入力する毎に予測文字は変化するため、多数の予測文字を並べると使用者に負担を強いることになる。予測文字の最大個数を 3~5 個程度にとどめ、この部分を除きアルファベット順配列の文字盤とすれば、使い易さをあまり損ねずに済むと思われる(図 1(b)(c))。

予測文字を文字盤左上にどのように置くかについては、いくつかの方法が考えられるが、使い易さを考慮すれば以下に示す方法が有力であろう。

(1) 予測文字の行配置

予測文字を並べて行を構成し先頭行に置く方法を、ここでは行配置と呼ぶ(図 1(b))。次文字が予測文字(図中斜線部)に無かった場合は、1回余分な操作が必要となる。予測文字の個数を増やしても損失とはならないので、効率だけを考えるならば可能な限り多く置けばよい。

(2) 予測文字の列配置

予測文字を1文字ずつスキャンする方法をここでは列配置と呼ぶ(図 1(c))。次文字が予測文字(図中斜線部)の筆頭に置かれていれば1回の操作で済むものの、予測文字から選ばれない場合は、予測なしの場合よりも予測文字の個数分だけ余分な操作が必要となる。従って、予測文字の個数を増やせば性能が向上するというわけではない。

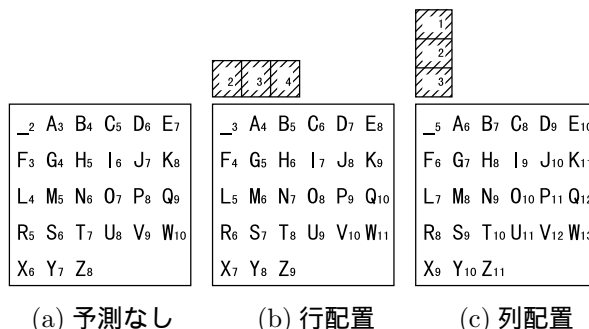


図 1: 予測文字(斜線部)の置き方(添え字は選択に要する操作回数)

2.2 予測文字の決め方

現在の文脈 s に基づく次文字 a の確率 $P(a|s)$ が得られているものとする¹。

行配置あるいは列配置される予測文字を決める方法のひとつに、図 1 斜線部の操作回数の少ない位置から順に確率の高い文字を置く方法があり、

¹ここでは、PPM 法 (Method C, blending[2]) を用いるが、別の方法で求めてもよい。

文献 [5] の方法はこの考え方に基づいている。

ここで、次のような場合について考えてみよう。図 1(b) で筆頭位置の予測文字 (2 回の操作で選択) を決めるに際し、次文字が「_」の確率が最も高く、「w」の確率がそれをごくわずか下回ったとする。それぞれ文字盤から選ばれた時の操作回数は 3 回と 11 回である。このような場合は、「平均的」には明らかに「w」を置くのが得策である。

以上のことを考慮し、ここでは、次式で計算される「次文字を選択するための操作回数の期待値」を小さくするように予測文字とその位置を 1 文字ずつ決めるという方法を用いた。

$$\sum_{a \in A} P(a|s) l_a \quad (1)$$

ここで、 l_a は文字盤から文字 a を選択するための操作回数である。

具体的には、行配置の場合も列配置の場合も、図の斜線部の操作回数の少ない位置から順に、以下の手順で置く文字を決める。ただし、予測文字の最大個数を m とする。

- (1) $i := 1$ とし、図 1(b) または (c) の斜線部を空とする。
- (2) 斜線部の i 番目の位置に置く文字として、式 (1) を最小にする文字 $a^{(i)}$ を求める。
- (3) $a^{(i)}$ を置くことにより期待値が下がらなければ終了する。
- (4) i 番目の位置に $a^{(i)}$ を置く。
- (5) $i := i + 1$ とし、 $i = m$ なら終了する。そうでなければ (2) に戻る。

上記 (2) で式 (1) の計算を行う際には、 l_a が固定されたものではないことに注意が必要である。つまり、行配置の場合は、文字盤から選択する文字の操作回数が初めにプラス 1 され、列配置の場合は、文字盤から選択する文字の操作回数が毎回プラス 1 されるということである。

また、上記の方法で予測文字を置くよりも全く置かない方がよい (式 (1) の期待値が小さい) と判断されるならば、予測文字を置かずに図 1(a) の文字盤のまま用いる。

行配置の場合、仮に次文字がほぼ正確に予測されるならば、複数の予測文字を行配置するよりも

「有力な 1 文字のみの行配置」とし、1 回の操作で選択できるようにするのが得策と考えられる。これは使用者のことを考慮しても、あまり不自然なことではない。この方法をここでは「行配置 +」として区別する。

表 1: 平均操作回数の比較

text (size)	book27 (729437)	book97 (763498)	gopc74 (851693)
予測なし	5.16	11.20	7.14
行配置	3.22	4.94	4.23
列配置	2.87	4.63	4.09
行配置 +	2.95	4.79	4.00

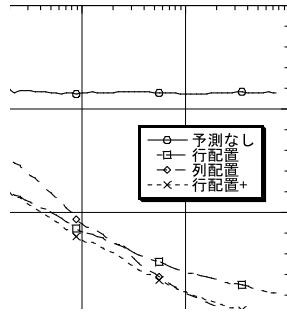
3 平均操作回数の比較と考察

予測文字の置き方が平均操作回数におよぼす影響をシミュレーションにより調べた。次文字の確率 $P(a|s)$ の計算には最大予測次数 $o = 5$ の PPM 法 (Method C, blending[2]) を用い、予測文字の最大個数を $m = 3$ とした。book27, book97, gopc74 の 3 つのテキスト (付録参照) について調べた結果を表 1 および図 2 に示す。

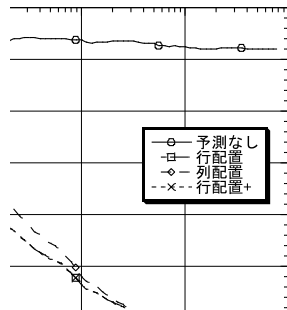
表 1 からは、最大 3 個の予測文字を置くだけで、操作回数が 40 - 60% 程度に軽減されることがわかる。置き方による違いは顕著ではないが、行配置に比べて列配置が優れていた。しかし、この表だけから性能を評価することは適当ではない。

図 2 は入力文字数による変化を示したものであるが、これら 3 つのグラフから共通の特徴を読み取ることができる。まず、予測文字の置き方によらず、100 文字程度の入力で予測文字を置いた効果が現れる。その後、ある程度入力テキストに対する学習が進むまで (30000 ~ 300000 文字程度まで) は、行配置が列配置よりも優れている。それ以後、列配置の性能が良くなり、表 1 の結果に至っていることがわかる。

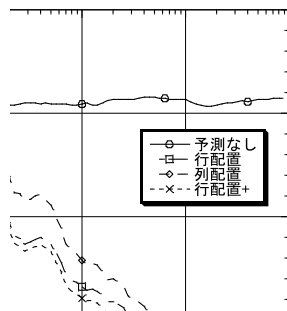
行配置 + に注目すると、1000 ~ 10000 文字程度までは行配置とほぼ変わらず、それ以後に違いが明瞭となる。十分な学習が行われた後には列配置が良いものの、それまでの広い範囲で、行配置 + が優れた性能を示した。これは、予測文字が 1 文



(a)book27



(b)book97



(c)gopc74

図 2: 平均操作回数の比較

字置かれた場合の予測が高い割合でヒットしていることを伺わせる。図 3 は、「予測文字を置いた

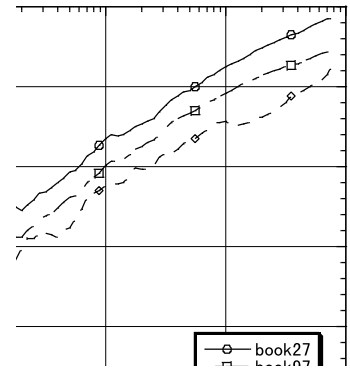


図 3: 1 文字目の予測文字のヒット率

場合に 1 文字目がヒットする割合」を示したものであるが、言語やアルファベットサイズによらず、最終的には 50% を超える高率で予測がヒットしており、行配置 行配置+ の性能の改善を裏付けている。

4 おわりに

現在の文脈における次文字の確率に基づき、次文字を選択する際の操作回数の期待値を求め、これを小さくする予測文字を置く方法を示した。また、予測文字の置き方が入力効率に及ぼす影響について検討した。

シミュレーションから、入力テキストについて十分な学習が行われた場合は、最大 3 個の予測文字を置くだけで、操作回数が 40 – 60% 程度に軽減されることがわかった。また、行配置に改良を加えた「行配置+」の方法は、広い範囲で良好な性能を示すことが確認された。

ここでは比較的大きな文字数のテキストを入力した場合について調べたが、今後は、所定のテキストを用いて事前に学習した場合の評価についても検討を行いたい。

参考文献

- [1] J.G. Cleary and I.H. Witten, "Data compression using adaptive coding and partial string matching," IEEE Trans. on Communication, Com-32, no.4, pp.396-402, 1984.
- [2] T.C. Bell, J.G. Cleary, and I.H. Witten, "Text compression," Prentice-Hall, 1990.
- [3] J.J. Darragh and I.H. Witten, "The Reactive Keyboard: A Predictive Typing Aid," IEEE Computer, vol.23, no.11, pp.41-49, 1990.
- [4] J.J. Darragh and I.H. Witten, "Adaptive Predictive Text Generation and the Reactive Keyboard," Interacting with Computers, vol.3, no.1, pp.27-50, 1991.
- [5] 小山智史, "文字予測機構を組み込んだ行列スキャン入力方式の検討," 信学技法 HCS98-28, pp.67-72, 1998.

付録 評価に用いたテキスト

3 の評価に用いたテキストを下表に示す ($|A|$ はアルファベットサイズ)。

テキスト名	$ A $	文字数	内容の一部
book27	27	729437	WHEN_FARMER_OAK_SMILED _THE_CORNERS_OF_HIS...
book97	97	763498	When Farmer Oak smiled, the corners of his...
gopc74	74	851693	トウシヨハ ヨムノモ ツク ルノモ マッキントツシュ...

book97 は, Calgary Corpus[2] の book1(英語の小説) から制御コマンドを削除したものを用了。book27 は book97 中の小文字を全て大文字に置換し, A-Z 以外の文字を「_」に置換し, 2 個以上連続する「_」を 1 個の「_」に置換したものである。gopc74 は, 原文(日本語の評論)をカナ文字列に直し, 文節ごとにスペースを入れたものを用了。これは, かな漢字変換のキーボード操作を模擬したものである。機械的に変換したもので, 一部不適切な変換結果も含まれているが, そのまま用了。