

3J-3

打鍵時間を利用した連文節かな漢字変換

椎尾一郎・金子宏

日本アイ・ビー・エム株式会社 東京基礎研究所

はじめに

日本語ワードプロセッサ利用者の打鍵が、文章の文節区切りで停滞することを、以前報告した。(1)

今回は、前回の報告より多くの被験者について、打鍵の時間間隔を測定した。これに基づいて、文節切り出しに、打鍵時間の情報を利用した、連文節かな漢字変換システムを試作して、その評価を行った。

打鍵の測定実験

前回の報告では、2人の被験者について、打鍵時間を測定した。今回は、個人差を測定する目的で、7人の被験者に対して次のような実験を行い、打鍵時間を測定した。

被験者 男性7名。当研究所社員。  
 打鍵内容 用意された15の短文から8-15文を複写。  
 入力方式 ローマ字入力。  
 測定方法 画面上部に漢字混じりの文を表示し、この下にかな文字で読みを打鍵してもらう。正しく打鍵された時の打鍵と時間を10msの単位で記録。

実験で得られたローマ字の打鍵データから、かなの区切りの場所の打鍵間隔に対して、以下の分析を行った。

図1は、被験者1が、15の短文のうちの一つを打鍵した時の打鍵間隔を示す。

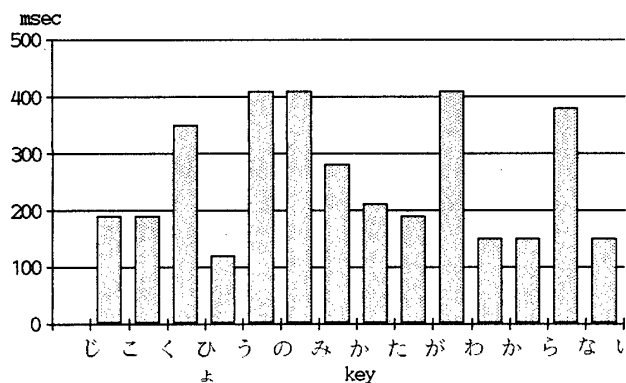


図1: 打鍵の例

これは、「時刻表の見方がわからない」という文を入力したときの打鍵間隔である。文節区切りの場所で打鍵間隔が大きくなっている。また、自立語と、付属語の間でも、停滞している事がわかる。7人の被験者の

打鍵の結果は、次のようになった。この表で、かな間とは、文節境界と、自立語-付属語境界の、両方を除いた場所での打鍵間隔である。

[被験者の打鍵間隔 (msec)]

被験者	1	2	3	4	5	6	7
かな間の平均	223	332	419	380	235	749	598
標準偏差	155	186	253	190	123	409	361
文節間の平均	500	732	801	542	262	1471	1051
標準偏差	233	498	396	165	140	419	529

かな漢字変換の試作

前節の測定結果をもとに、打鍵時間情報を利用した、かな漢字変換を試作した。これは、従来のかな漢字変換システム(PS/55日本語連文節変換プログラム)の、変換アルゴリズムに、打鍵情報を利用する機能を付加したものである。本システムが、打鍵情報を処理し、文節を切り出す方法を、以下に説明する。

文節切り奨励点の算出

前回の報告によると、使用者の打鍵は、文節区切りで、必ずしも停滞するわけではない。一方、それ以外の場所で、打鍵が停滞する事は少なく、比較的安定している。よって、前出の、かな間の時間を基準として、打鍵間隔を評価するのが適切である。

筆者等は、打鍵間隔  $t_k$  が、かな間の平均時間より長い場合、この場所が文節区切りである可能性があり、その程度は、標準偏差に反比例すると考えた。そこで、 $t \geq \mu_k$  である場合に、文節区切りとなることを奨励する奨励点  $E$ 、

$$E = A \times \frac{(t - \mu_k)}{\sigma_k} \quad (1)$$

を、考えることにした。ここで、 $\mu_k$ は、かな間打鍵間隔の平均、 $\sigma_k$ は、その標準偏差、 $A$ は、正の定数である。

$\mu_k$ と、 $\sigma_k$ は、使用者の個人差や、習熟度により異なるので、使用者の履歴から測定する。 $\mu_k$ の測定は、次のように行う。使用者が、変換キーや句読点で区切られた句を入力する度に、その句におけるかな間の打鍵間隔の平均を求め、 $n$ 番目の句における、かな間の打鍵間隔の平均を、 $\bar{k}(n)$ とする。 $n+1$ 番目の句で使用すべきかな間の平均  $\mu_k(n+1)$ の値を、

$$\mu_k(n+1) = w \times \bar{k}(n) + (1-w) \times \mu_k(n) \quad (2)$$

により計算する。ただし、 $w$ は、1以下の正の定数である。

$\sigma_k$ は、次のように測定する。 $n$ 番目の句における、かな間の打鍵間隔の自乗の平均を、 $k^2(n)$ とする。 $n+1$ 番目の句での、かな間の自乗の平均  $M_k(n+1)$ を、(2)式と同様に、

$$M_k(n+1) = w \times \overline{k^2}(n) + (1-w) \times M_k(n) \quad (3)$$

として求める。これから、

$$\sigma_k(n+1) = \sqrt{M_k(n+1) - (\mu_k(n+1))^2} \quad (4)$$

により、 $n+1$ 番目の句で使用すべき標準偏差  $\sigma_k(n+1)$ を計算する。

(2),(3)式における $w$ の値を大きくすると、句ごとの使用者のちょっとした揺らぎが、次の句のしきい値に影響してしまう。逆に、 $w$ の値を小さく設定すると、ある使用者が作業を始めてから、その使用者の特性を把握するまでに、 $w$ の逆数程度の数の句の情報が必要になる。実験から、 $\mu_k + \sigma_k$ 、の値の変動が20ms程度になる値を捜した結果、 $w = 0.1$ とした。

#### 文節切り出しの方法

試作の土台になった、かな漢字変換は、文節数最小法を基本にしている。原則として、自立語に罰点 10、接辞に罰点 6、通常の助詞に罰点 0、希に出現する助詞に罰点 2が与えられている。句全体で、この罰点の合計が最小になるように、文節の切り出しが行われる。

この処理に加えて、打鍵時間の情報が、次のように利用される。文節区切りの場所での、式(1)による奨励点が、句全体の罰点から、減らされるのである。この結果、停滞時間の長い場所を文節区切りとしたとき、罰点が軽減されることになる。

$\mu_k + \sigma_k$ の停滞で、希に出現する助詞の罰点を打ち消す程度の、奨励点を与えることにした。そこで、式(1)の定数  $A = 3$ とした。

#### 自立語—付属語境界の考慮

使用者の打鍵は、文節以外の場所でも、停滞することがある。とくに、図 1に見られるように、自立語と付属語の間で、停滞することがある。このことが、誤った文節区切りを誘発することが考えられる。そこで、自立語と付属語の間でも、打鍵時間を考慮することにした。具体的には、自立語—付属語間の奨励点  $E$ の2分の1を、句全体の罰点から差し引いている。

文節区切りと、自立語—付属語境界の他では、打鍵時間を考慮していない。この結果、文節の区切りになる可能性が小さい場所での不用意な停滞は、結果に悪影響を及ぼし

にくい。このような場所を、文節区切りにしようとする、文法的な解析によって、大きな罰点が与えられるからである。

#### 動作試験

ここで、試作したかな漢字変換を評価する目的で、先の実験の被験者に、本方式を説明して、同じ短文を、もう一度打鍵してもらった。実験に用いた15の短文の中には、改造前のかな漢字変換で、文節区切りの失敗を起こす文が、6文含まれている。例えば、図 1で例示した文は、

「時刻表のみ片側からない」

と変換されてしまう。被験者は、このうち5ないし6文を打鍵している。打鍵時間を考慮する効果により、このうち一部の文は、正しく変換された。一方、もともと正しく変換される文が、打鍵間隔の停滞<sup>9</sup>より、誤変換されるという副作用もあった。次の表は、この効果と副作用の集計である。

[本方式の効果と副作用 (単位: 文)]

被験者	1	2	3	4	5	6	7
打鍵した文	15	8	9	9	15	15	15
誤変換される文	6	5	5	5	6	6	6
効果	4	3	4	5	6	4	4
正しく変換される文	9	3	4	4	9	9	9
副作用	0	0	0	1	1	2	0

この実験例では、被験者全員の変換率が改善された。一方、従来方式で、文節区切り間違いにより誤変換される文の比率は、20%程度である。これが実験の設定よりも低いことを考慮すると、被験者4と6にとっては、副作用も大きいと考えられる。

今回の実験は、一短文を複写する作業であった。長い文の複写や、文の産出作業の場合には、さらに効果が上がると考えている。

#### まとめと今後の方針

日本語入力する使用者の打鍵を測定した。これをもとに、打鍵時間の情報を文節区切りに利用した、かな漢字変換を試作した。これによる、変換率の改善の効果を確認した。

今後、長文の複写と産出の打鍵データを用いて、本方式の効果測定する予定である。

#### 文献

1. 椎尾一郎: 思考時間に基づくユーザインタフェース, 情報処理学会第35回全国大会 7E-1 (1986) pp. 2559-2560.