ヒューマンインタフェース 47−17
（1993. 3. 2）

読み情報をもつ日本語エディタの作成と評価

畠山 勉　角田 博保
電気通信大学 情報工学科

かな漢字変換型ワープロの利用者は、漢字を入力する際、漢字かな混じりの表記を読みで指定する。この読みを捨てずに保持し利用すれば、漸増探索指令や動的展開を、英文に対する時と同じ感覚で実行でき、かな漢字変換のやり直し(再変換)の効率も増す。本研究ではこの機能をもつエディタ(JEM)を漢字かな混じりテキストを保持するバッファ(漢字層)と読み情報を捨てずに保持するバッファ(読み層)の多層構造で実現することを考え、GNU Emacs系のエディタであるNepoch上に上位互換に近い形で実装した。機能評価実験を行なったところ、JEMは漸増探索指令において従来の方法より有利とわかった。

The design and evaluation of a Japanese Editor with pronunciation-information

Tsutomu Hatakeyama　Hiroyasu Kakuda
Department of Computer Science, The University of Electro-Communications

When one uses a word-processor of Kana-to-Kanji conversion type, he/she specifies a sentence with its pronunciation. If an editor preserves and uses these pronunciation-information, a user can use I-search(incremental search) or dynamic abbreviate expansion in Japanese sentences as same as in English ones, and can perform Kana-to-Kanji re-conversion efficiently. So the authors have designed and developed an editor, called "JEM", with multi-layered-structure of Kanji-layer (contains normal text) and Yomi-layer (contains pronunciation-information). JEM is implemented on Nepoch that is one of the GNU Emacs series. The authors have evaluated its effectiveness through experiments on I-search.

1 はじめに

日本語の文章を作成するのに かな漢字変換が広く利用されている。かな漢字変換では、利用者は 漢字かな混じりの表記を直接指定することはできないので、 読みを入力して、 適当な文節に区切ったり、複数の変換候補から一つを選ぶ作業を経て、目的の漢字かな混じり文を得る。 そして、基となった読みは捨ててしまう。

一方で、利用者はテキスト内での移動や探索のためにしばしば**逐次探索指令** (incremental search. 以下、**漸増探索**と略す) を使う。しかし日本語を含む文章で日本語の部分に対して、それを行なうのは、変換が成功しそうな、ある程度の文字数まで入力して漢字に変換する作業が必要である。そのため、ある程度以上離れた位置への探索ならばともかく、ちょっとした移動のために英文のときと同じ頻度で使われることはない。また、プログラムなど、同じ名前が頻繁に登場するテキストに対して特に有効な**動的展開**による入力についても、やはり、変換が成功しそうな、ある程度の文字数まで入力して漢字に変換する作業が必要なので、英文のときと同じ頻度で使われることはない。

しかし漢字かな混じりの表記のレベルで照合させようとするのではなく、読みのレベルで照合させるならば、変換が成功しそうなところまで入力する必要はなくなる。

それには漢字の入力の時に、いつも捨てられていた読みと文節区切り (合わせて**読み情報**と呼ぶ) を保持する機構が必要になる。本研究では漢字かな混じりテキストを保持するバッファ (漢字層) と読み情報を捨てずに保持するバッファ(読み層) の多層構造を持ったエディタを、GNU Emacs 系のエディタである Nepoch 上に実装した。これを **JEM** (Japanese Editor with Multi-layered Structure) と名付ける。

2 JEM V2 の設計

本節では JEM の外部仕様について述べる。

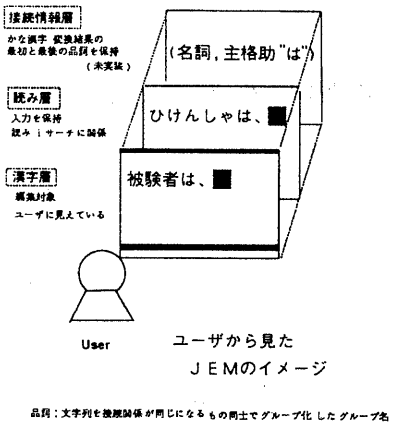

図 1: ユーザからみた JEM のイメージ

2.1 多層構造

2.1.1 ユーザから見た多層構造

ユーザから見ると、JEM は次のように振舞わなければいけない (図 1) 。
(1) 各文節の入力時の読みを理解している。
(2) 各層のカーソルが同期して動いている。
(3) 漢字層の変更は 各層に反映されている。

2.2 漢字層と読み層

漢字層は 利用者が普段 編集対象としているバッファのことである。そして、漢字層の読み情報を利用できる形で保持するバッファを**読み層**とよぶ。読み層に関連づけられたファイル(読み層ファイル) は漢字層に関連づけられたファイルの名前に拡張子 ".yomi" を付加した名前がつけられる。もし読み層ファイルが存在しない場合は、**Wnn** の逆変換を用いて漢字層ファイルから自動生成される。

2.3 削除・挿入の扱い

一方の層のオフセットに両層の対応する文字列の長さの比率を掛けて四捨五入した値を他方の層のオフセットとすることで、両層の対応づけを決定する (図 2)。例で、漢字層の "第" を削除すると、読み層の "だい" が同期して削除される。漢字層の "1" と "回" の間に文字列を

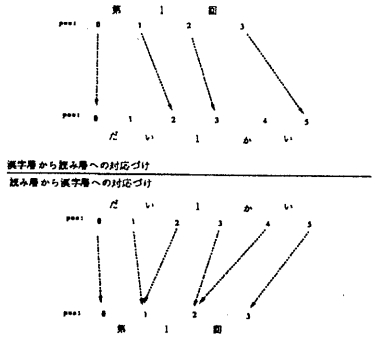

図 2: 両層の対応づけ

挿入すると、それは読み層の "1" と "かい" の間に同期して挿入される。[1]

2.4 読みによる漸増探索

英字版と同じ使用感を得ることを目標としている。

2.4.1 読みによる漸増探索の外部仕様

"C-s" に続けて読みをローマ字で打鍵すると順方向の探索が始まる。ローマ字はかな漢字変換され、探索文字列の平仮名が確定するたびに探索が実行されることが漸増探索の最大の特徴である。

逆方向に探索したければ "C-r" で起動するが、順方向探索を起動した後、いつでも逆方向探索に切換えることができる。全角半角は区別しない。また、探索文字列は通常文節の先頭にあたる文字から入力されるのが普通であるので、文節先頭からのみ照合するオプションを設けている。複数行に渡る探索文字列については、文字 と 文字 の間に 0 または 1 個の改行コードが入っても照合するので、改行を意識しないで探索できる。入力のモードをトグルするキーで切換えて入力することで、探索文字列にアルファベットを加えることが可能である。探索文字列に英字を含む場合、大文字小文字の区別することも可能である。正規表現は指定することはできない。

[1] 図の例は、たまたま両層の対応づけがうまくいっている場合だが、「第 1 位」と「だい 1 い」の時は、"第", "1", "位" に対し、 "だ", "い 1", "い"、が対応づけられるので、漢字層の "1" を削除すると読み層の "い 1" が削除されてしまう。

2.5 再変換

仮名漢字変換の利用者はしばしば思考の流れを中断したくないなどの理由で大して確認もせずに確定し、どんどん先に進む。そのようなとき、後で直す作業が必要になる。最も単純な直し方は誤った部分を削除して再度入力することである。しかし、変換結果は望んだものでなくても、その読みは望んだものと同じ読みなのであるから、読みだけ再利用すれば、再び打ち込む必要がない。読みを再利用して変換することを**再変換**と呼ぶことにする。

再変換は読み情報を使わなくても可能であるが、その場合、精度が高くないのが難点である。しかし JEM では入力時に保存した読み情報を利用して再変換することができる。

2.6 読みによる動的展開

動的展開 (dynamic abbreviate expansion) は、現在入力中の単語 (文節) と同じ表記で始まる単語を現在編集中のテキストなどから探して、補完する入力方法である。

動的展開はプログラムなど、同じ単語を頻繁に使うテキストの編集で特に有効である。

日本文では漢字を含む語を動的展開しようという場合、入力した文字列 (ローマ字ないし平仮名) を漢字に変換しなければ対応づけることができない。しかしながら JEM は読みを保持しているので、そのまま読み層において略称と展開結果の対応づけを行ない、展開結果に相当する位置の漢字層文字列を展開結果とすることで動的展開をごく自然に実現している。

JEM での読みによる動的展開は日本語入力モードで起動する。日本語入力モードで文字列の先頭の方を打ち込んでから "M-/" を打鍵すると読みによる動的展開が起こる。

展開の次候補を得るには「スペースバー」または「メタキーを押しながら /」を打鍵する。現在ポイントから 逆方向に 順に探される。バッファの先頭まで行ったら、現在ポイントから 順方向に順に探される。なんらかの、次にやりた

いコマンドをやれば確定されてそのコマンドが実行される。

また、新たな展開候補となるための条件は、「これまでの全ての展開候補と漢字部分が異なること」としている。なお、入力はローマ字仮名変換の途中までで良い。(例 “き sh” は「“きしゃ” または “きしゅ” または “きしょ”」として展開される。)

3 JEM V2 の実装

JEM の試作版[2]では文字単位の編集ができない(文節単位のみ)、漸増探索が高速でないという欠点があった。それらの反省を踏まえて実装した。

3.1 実装の基本方針

複数のバッファによる多層構造を用いて、読み情報・文節区切り情報を保持する。これは、編集対象のバッファ(漢字層バッファ)を読み情報を保持しているバッファ(読み層バッファ)と関連づけることで行なう。

また、計算機資源との兼ね合いとして、プログラムサイズが大きくなったり、メモリを多く消費することがあっても、原則として高速化と気のきいたユーザインタフェースのほうを優先する。

3.2 JEM V2 のシステム構成

JEM は X11 R5 の下で Nepoch 1.1[2] に手を加えたものとして実現されている。日本語環境としては Wnn V4 を、日本語フロントエンドプロセッサには EGG (たかな) を用いている。

JEM の部分のソースコードは Emacs Lisp で現在約 1 万行余りである。そのうち、読みによる漸増探索の部分は 約1650 行、動的展開の部分は 約1200 行である。図3に JEM V2 のシステム構成を示す。

[2]JEM の母体となるエディタは、Nepoch の 第1.1 版を使用した。それは、GNU Emacs の X11 環境下の使用を前提にして、X11 のサービスを利用できるように GNU Emacs に対しいくつかの拡張を行なった epoch の 第3.2 版の日本語版である。epoch は GNU Emacs と互換性を持ち、従来の Emacs Lisp コードはそのまま動作する。

特徴としては、X11 環境のマルチウィンドウのサポート, X11 の描画属性を使ったテキストリージョンのマーキング (color, underline, stipple, font) などが挙げられる。

Nepoch は広く使われており、ソースコードの入手・改変・再配布が自由である。今回 JEM を Nepoch 上に実装することにしたのは、文節区切りが表示できるからである。文節単位で編集する場合、文節区切りが表示されないと、次の予想がつかず使いにくい[2]。epoch の表示属性機能を使わないのであれば、JEM は X Window のドでなくても実現可能である。

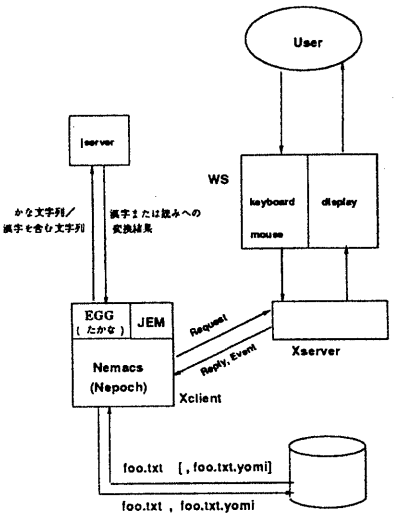

図 3: JEM のシステム構成

3.3 読み情報などの持たせ方

読み層の 第 i 行を 漢字層の 第 i 行に対応させる。

(1) (**読み層バッファ**) ::= (読み情報の行)*
(2) (**読み情報の行**) ::= {(文節レコード)}*(改行文字 | バッファの終り)
(3) (**文節レコード**) ::= (レコード先頭を表す 2 バイト文字)(漢字層での長さを表す 2 バイト文字 2 個)(読み方情報)(レコード末尾を表す 2 バイト文字)
(4) (**読み方情報**) ::= (ひらがな | 記号)*

ここで (**読み方情報**) は日本語フロントエンドプロセッサを介していない時の自己挿入文字のときは、そのものの 1 文字以上の連接であり、そうでないときはローマ字仮名変換の結果 (ひらがなや句読点などの 1 文字以上の連接) である。

3.3.1 文節レコード

文節レコードと読み方文字列について。

1 文節の、JEM の読み層における表現を**文節レコード**とよぶことにする。その文節の読み情報を持っている。文節レコードは次の構成要素からなる。

(1) 漢字層での漢字表記 (文字列)
(2) 読み方文字列 (文字列)
(3) 漢字層での漢字表記の長さ (バイト単位の非負整数)

漢字層が次の 3 行からなるテキストを考える。

```
彼は新しい
gnus.el を
インストールしました。
```

このとき、読み情報として

4	かれは 6 あたらしい
5	gnus 5 el を
22	いんすとーるしました。

を持たせる。

3.4 エディタ (Nepoch) 本体と JEM モジュールとのインタフェース

3.4.1 読み層の同期の取り方

epoch のコンパイル時に "HAVE_LEIF" オプションを指定して得られる、before-change-function と after-change-function の 2 つの hook に、漢字層の変更を読み層に反映させる hook (jem-synchronous-hook) を登録し、それが呼ばれることで実現する。

jem-synchronous-hook は漢字層の変更が「削除」ならば 読み層の対応リージョンを計算して削除を行ない、「挿入」ならば逆変換を行なって読み層の対応ポイントに挿入する。

4 JEM の評価実験

読みによる漸増探索について、評価実験を行なった。

4.1 実験 1 読みによる漸増探索と 変換型探索の比較実験

(実験 1h) 変換型探索, (実験 1y) 読みによる漸増探索.
キーストローク数からは読みによる漸増探索のほうが速いはずだが、実際に確かめることが目的。

4.1.1 実験に使うコマンドの外部仕様

読み 漸増探索の外部仕様は別項の通りである。また、変換型 漸増探索は Wnn+EGG の利用者に一般に良く使われているものを多少手直ししたものである。それは C-s で変換型 漸増探索に入り、C-k に続いて探索文字列として漢字を入力することができる。(ミニバッファを改めて日本語入力状態にするキーは不要にしてある。漢字入力には EGG を用いている。) ミニバッファの変換を確定した後、リターンキーで探索が起こる。

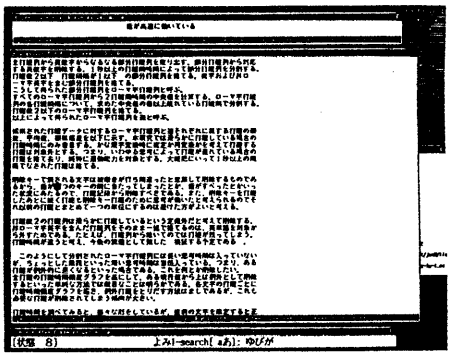

図 4: 「指が高速に動いている」を探している途中

4.1.2 方法

被験者 被験者は、筆者らが所属するの研究室の 9 名 (男性 6 名・女性 3 名) で、年齢は全員 22 歳前後である。全員とも普段から Emacs を使用しているが、他のエディタの方を使い慣れている被験者がいた。また、通常は漸増探索を使うことは滅多にないという被験者もいた。

原稿 英字と、句読点以外の記号を削除し、さらに改行コード以外はすべて全角文字に変換した 750 行分, 各行 70 カラム (35 字) の日本文のテキストファイルを用意した。題材は ヒューマンインタフェース関係の論文 (4 編) からとった。問題文の読み層ファイルは逆変換で生成したものを入力時の読みとなるように、手動で補正した。

ターゲットの文字列

必ず文節の先頭から始まる。平均長 10.76 文字 (標準偏差 2.56 文字) 文字列の開始位置は行頭・行末の 5 文字分を外して乱数で決めた。

ワークステーション Sun Sparc Station SLC と ELC を併用した。パフォーマンスの差が問題になるようなタスクではない.

ディスプレイ 17 inch, グレースケールディスプレイ。フォントは本文の部分と問題表次窓の部分には **a18** と **kanji18** を、ミニバッファの部分には **12x24** と **k24** を使用した。

本文用のウインドウのサイズは 112x36 に統一し (バッファには 36 行、114 カラム表示される)、ミニバッファはその下方においた。本文用のウインドウの上に問題文用のウインドウを揃えておいた (図 4)。

収集されるデータ　キーとその押された時刻 (精度は 30 msec 程度), キーシーケンスから決定されたコマンド名, そのコマンドが実行される前の 画面の状態に関するデータ (画面の開始ポイント値, 現在ポイント値, 画面内での行番号, 現在ポイントのカラム値, 現在行のカラム数, ポイントの前/後の文字), バッファ名

手続き

(1) **練習**　被験者は十分な練習をしてから実験に入った。特に変換型の探索をよく練習してもらった。

(2) **課題**　原稿修正タスクを模擬した次のタスクを課す。画面上にミニバッファとは別に横に細長い窓 (問題窓) が上部に置かれ、そこに問題として「行くべき場所」を指示される。その場所まで、(変換型漸増探索・読みによる漸増探索) のいずれか 1 つを 主に使用して、必要ならば基本コマンドを補助的に使いながら、指示された場所へ移動する。そこで削除キー C-d (コントロールキーを押しながら d) または C-h を押して正しい答ならば正しい答のリージョンに下線がつき (「正解」の意)、次の問題に進むことができる。

(3) **正解基準**　問題の文字列上とその前後 1 文字の範囲とした。

(4) **問題数**　問題は 1 セッションに 50 問であり、被験者は全問を順に解かなければいけない。実際の所要時間は 10 分程度であった。

4.1.3　実験 1 の結果

C-s を押してから問題を獲得するか獲得してしまってから C-s を押すかを特定できないので、こうした。第 n 問の正解部分に対する C-d (または C-h.) から次の問題の正解部分に対する C-d (または C-h.) までを「第 n 問にかかった時間」に着目する。データは C-s C-k の C-k の分を補正していないままの値である。得られたデータとそのサマリを巻末の表 1 と 2 に示す。ただし下記の除去基準が施されている。

次のデータは捨てた

(1) 1 セッション (50 問) の初めの 5 問

(2) サーチ中に C-g または DEL を打鍵したもの (ミス直し) とコマンド "abort-recursive-edit" のキーが打鍵された問題。

(3) ターゲットが順方向にない問題 (約半数) と C-r を打鍵したもの。

(4) 差が 20 秒 (20000 msec) よりも大きかった問題 (4 件)。これは被験者が実験者にやりかたについての質問をしていたときなどがこれに当るからである。

4.1.4　実験 1 の考察

有効な問題数は 7 から 19 しか残らなかったが、1 問当り 1.7 秒から 6.5 秒の範囲で読み漸増探索の方が速い (分散は 3.0 から 49.9)。変換型は変換のため以外に不要な打鍵が 3 打鍵ほどあるが、これを 1 問当り 0.6 秒として補正してもやはり読み漸増探索の方が速い。変換のための打鍵が必要か不要かの違いがそのまま現れた結果といえる。

4.2　実験 2 読み漸増探索による移動とマウスによる移動の比較実験

(実験 2m) マウス型 (実験 2y) 読み型。

ターゲットが画面内にある場合、打鍵レベル模型 [3] からはマウスによる移動方法が有利と推測されるが、それが読み漸増探索と比べてどのくらい有利なのかを調べる。

4.2.1　実験に使うコマンドの外部仕様

読み 漸増探索の外部仕様は別項の通りである。マウスによる移動はマウスの左ボタンを押すと マウスカーソル (鉛筆の形をしていて大きさ約 1cm) の位置に、カーソルが移動する。

4.2.2 方法

被験者 実験1に同じ。

原稿 原稿は実験1に同じであるが、実験1では問題を画面内に問題窓として提示したのに対し、実験2ではハードコピーにオレンジ色の蛍光ペンを施したものとして提示する。

ターゲットの文字列 実験1と同様の基準で新たに選び直した。平均長 11.32 文字 (標準偏差 2.96 文字) である。

ワークステーション・ディスプレイ・収集されるデータ 実験1に同じ

手続き

(1) **練習** 被験者は十分な練習をしてから実験に入った。特に変換型の探索をよく練習してもらった。

(2) **課題** 原稿修正タスクを模擬した次のタスクを課す。ハードコピーに示された問題 (50 問で行番号順に整列されている) を「行くべき場所」を指示される。その場所まで、(マウスによる移動・読み漸増探索による移動) のいずれか1つを 主に使用して、必要ならば基本コマンドを補助的に使いながら、指示された場所へ移動する。そこで削除キー C-d (コントロールキーを押しながら d) または C-h を押して正しい答ならば正しい答のリージョンに下線がつき (「正解」の意)、次の問題に進むことができる。

(3) **正解基準・問題数** 実験1に同じ

4.2.3 実験2の結果

実験1と同様に第 n 問の正解部分に対する C-d (または C-h.) から次の問題の正解部分に対する C-d (または C-h.) までを「第 n 問にかかった時間」に着目する。得られたデータを巻末の表 3 に示す。ただし下記の除去基準が施されている。

次のデータは捨てたサーチ中に C-g または DEL を打鍵したもの (ミス直し) のキーが打鍵された問題。

4.2.4 実験2の考察

画面内にターゲットがあるならば、打鍵レベル模型の示す通り、ほとんどの場合マウスのほうが速かった。

しかし、個人差はあるものの、距離 (次の問題のターゲットまでの行数) が 10 を超えると、ターゲットが画面内に入っていても両者の差があまりないことが読みとれる。

5 今後の課題

今後の課題としてはまず、実験データの精密な解析と再実験が必要だと思う。また、システムの実装面では、品詞情報の保持と、標準にはありながら、JEM では削られている narrowing 機能を実現すべきである。

さらに主要なメジャーモードの JEM 化として、Jem LaTeXMode (LaTeXMode と JEM Mode を共存させたもの), Jem 日本語 C Mode を考えている。後者はプログラムの識別子として、動的展開が多用されることが考えられるので、有力である。

読みによる動的展開は現在の外部仕様がベストであるとは思えない。今後、十分な使用を経て次の点について、より妥当な仕様を探すべきである (おくりがな や 助詞も展開結果に付随するのが良いか悪いか, 展開結果の文節の長さ (現在は 1), 展開結果があまりに短い時は展開しないのが良いか悪いか, 前の候補に戻ることはできないのは良いか悪いか, など)

参考文献

[1] 角田博保: 多層テキスト構造を持つ構造エディタ, 第 27 回プログラミング シンポジウム (1986.1)

[2] 北村譲治: 多層構造の日本語エディタの試作, 平成 3 年度電気通信大学 情報工学科 卒業研究

[3] Card,S.K. , Moran,T.P. and Newell,A. The Psychology of Human Computer Interaction, *Lawrence Erlbaum Associates, 1983*

表1. 実験1の結果

問題毎の所要時間(単位 秒)　｜変換型(x), 読み型(y), 差(x-y)｜

問題	MA x	MA y	MA x-y	KS x	KS y	KS x-y	FT x	FT y	FT x-y	IK x	IK y	IK x-y	HM x	HM y	HM x-y	KM x	KM y	KM x-y	NT x	NT y	NT x-y	SK x	SK y	SK x-y	TY x	TY y	TY x-y	問題文
Q-007	10.8	6.9	3.9	19.0	8.2	10.8	9.4	11.5	-2.1	12.4	6.3	6.1	11.3	11.4	-0.0	8.7	8.3	0.4	8.6	12.0	-3.4	-	-	-	28.6	10.8	17.9	英数字を削除する
Q-009	11.6	19.1	-7.4	14.3	6.2	8.1	7.6	10.9	-3.4	9.8	5.9	3.9	17.0	6.0	11.0	-	-	-	10.4	6.8	3.7	13.1	8.5	4.7	14.0	9.4	4.7	単位とほぼ一致していた
Q-011	15.3	6.7	8.6	-	-	-	18.1	9.0	9.1	17.2	9.5	7.7	18.5	8.4	10.1	11.9	7.6	4.3	12.8	14.2	-1.4	-	-	-	17.3	10.9	6.4	局所的運動動作を測定するため
Q-012	15.4	6.3	9.1	13.9	7.8	6.1	7.1	7.4	-0.3	-	-	-	-	-	-	9.7	7.1	2.6	9.7	8.0	1.6	-	-	-	14.8	7.7	7.1	取り扱うことにする
Q-014	9.5	9.0	0.4	-	-	-	11.1	6.9	4.1	-	-	-	10.6	7.4	3.2	8.6	8.4	0.3	13.7	10.1	3.6	9.9	11.2	-1.3	-	-	-	同等の速度で打つ習慣
Q-015	9.4	8.4	1.0	13.4	7.5	5.8	-	-	-	12.2	4.6	7.6	9.3	10.2	-1.0	9.9	8.7	1.2	-	-	-	-	-	-	-	-	-	打鍵についてグラフ
Q-017	-	-	-	21.6	7.2	14.4	8.9	9.6	-0.8	12.6	6.0	6.5	21.4	13.1	8.2	9.7	6.9	2.7	15.1	10.2	4.9	13.5	10.3	3.2	25.2	15.7	9.5	中央値どうしでの比較
Q-018	22.7	7.0	15.7	19.4	8.7	10.8	-	-	-	-	-	-	-	-	-	9.9	8.2	1.7	-	-	-	-	-	-	-	-	-	統計値によって例外打鍵を削除
Q-021	-	-	-	13.1	7.9	5.2	10.1	8.0	2.0	9.3	2.9	6.4	14.8	5.5	9.3	6.5	6.3	0.2	15.3	5.8	9.5	-	-	-	19.1	16.2	2.9	速を計算した結果
Q-024	15.0	12.0	3.0	15.6	14.5	1.1	11.1	14.5	-3.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	両手打鍵の高速なもの
Q-027	-	-	-	21.1	12.2	9.0	-	-	-	-	-	-	11.9	14.5	-2.6	-	-	-	29.6	16.7	12.9	-	-	-	-	-	-	分かちの方がなめらかに打鍵
Q-028	19.5	5.4	14.1	11.8	6.3	5.5	-	-	-	-	-	-	-	-	-	-	-	-	11.9	9.6	2.4	-	-	-	-	-	-	事象サーバに転送
Q-029	7.2	6.0	1.2	10.4	5.5	5.0	6.9	7.3	-0.4	7.2	4.3	2.9	9.9	7.2	2.7	9.5	6.1	3.4	11.5	6.2	5.4	24.4	8.3	16.1	13.8	9.2	4.6	きれいに削除された
Q-031	7.7	9.8	-2.1	12.2	7.0	5.2	9.2	6.9	2.3	10.9	5.6	5.3	9.6	6.3	3.4	-	-	-	8.8	8.8	-0.0	9.8	7.2	2.7	14.7	10.5	4.2	さらに検討を行ない
Q-036	-	-	-	16.0	8.8	7.2	8.2	6.9	1.3	9.1	6.1	3.0	12.8	10.9	1.9	10.5	8.2	2.3	13.3	9.8	3.5	-	-	-	18.4	9.8	8.6	文章内容を考える時間も
Q-037	-	-	-	8.9	7.9	1.1	9.7	6.0	3.8	-	-	-	8.6	5.6	3.0	-	-	-	-	-	-	-	-	-	14.1	12.6	1.5	一般的である
Q-039	15.3	6.5	8.8	16.6	10.4	6.2	8.2	6.8	1.4	-	-	-	11.7	8.4	3.3	-	-	-	20.6	11.9	8.7	9.9	12.0	-2.1	-	-	-	作文打ちすることができない
Q-040	10.7	5.9	4.7	22.3	9.7	12.6	-	-	-	13.7	7.1	6.6	9.7	7.0	2.6	16.4	10.0	6.4	-	-	-	-	-	-	14.5	11.7	2.8	実際に論文を書くといった
Q-044	10.6	13.8	-3.1	12.7	11.3	1.4	17.8	7.3	10.4	-	-	-	18.4	11.0	7.3	8.6	14.0	-5.4	-	-	-	-	-	-	13.1	13.7	-0.6	同じテキストを打った場合
Q-046	-	-	-	14.6	7.3	7.3	11.7	5.3	6.4	10.0	5.8	4.3	-	-	-	8.4	6.1	2.3	12.5	7.8	4.7	24.2	10.3	13.8	-	-	-	句読点で区切られた文
Q-047	17.9	6.4	11.5	7.5	6.6	0.9	8.4	4.9	3.5	-	-	-	6.1	6.5	-0.5	8.0	7.1	0.9	-	-	-	-	-	-	12.0	9.6	2.4	大切である
Q-050	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	高速に打鍵できることがわかった

表2　実験1の結果（サマリ）

被験者(有効問題数)	MA(15) x	MA(15) y	MA(15) x-y	KS(19) x	KS(19) y	KS(19) x-y	FT(16) x	FT(16) y	FT(16) x-y	IK(11) x	IK(11) y	IK(11) x-y	HM(16) x	HM(16) y	HM(16) x-y	KM(14) x	KM(14) y	KM(14) x-y	NT(14) x	NT(14) y	NT(14) x-y	SK(7) x	SK(7) y	SK(7) x-y	TY(13) x	TY(13) y	TY(13) x-y
総時間	198.68	129.15	69.53	284.70	160.87	123.83	163.29	129.42	33.87	124.33	64.06	60.27	201.36	139.44	61.92	136.32	113.08	23.24	193.99	137.91	56.08	104.86	67.78	37.08	219.62	147.76	71.86
平均値	13.24	8.61	4.64	14.98	8.47	6.52	10.21	8.09	2.12	11.30	5.82	5.48	12.59	8.71	3.87	9.74	8.08	1.66	13.86	9.85	4.01	14.98	9.68	5.30	16.89	11.37	5.53
分散値	20.30	14.17	43.65	17.73	5.14	14.99	11.03	6.28	16.26	7.49	2.84	3.02	18.00	7.87	17.13	5.27	4.12	7.06	30.15	9.45	18.69	42.67	3.05	49.89	24.42	6.51	21.83
最大値	22.68	19.08	15.72	22.30	14.49	14.38	18.06	14.46	10.41	17.25	9.51	7.74	21.36	14.50	10.95	16.39	13.98	6.40	29.61	16.68	12.93	24.36	12.03	16.08	28.65	16.17	17.90
最小値	7.17	5.40	-7.44	7.50	5.46	0.87	6.90	4.89	-3.39	7.20	2.85	2.91	6.06	5.46	-2.62	6.51	6.09	-5.43	8.64	5.85	-3.36	9.84	7.17	-2.10	12.00	7.68	-0.57

表3　実験2の結果

"|" で区切られた各レコードの、初めの２つの欄 (0 または 1) は、
マウス(実験2m), 読み漸増探索(実験2y) それぞれについて、問題の場所が 見えている(1), 見えていない(0) を表す。
その各レコードの第３欄はマウス(実験2m)にかかった時間(単位 100 msec)
第４欄は読み漸増探索(実験2y)にかかった時間(単位 100 msec)
第５欄は第３，４欄の差を表している。

距離	MA 表示	MA m	MA y	MA 差	KS 表示	KS m	KS y	KS 差	FT 表示	FT m	FT y	FT 差	IK 表示	IK m	IK y	IK 差	HM 表示	HM m	HM y	HM 差	KM 表示	KM m	KM y	KM 差	NT 表示	NT m	NT y	NT 差	SK 表示	SK m	SK y	SK 差	TY 表示	TY m	TY y	TY 差	問題番号、問題文
1	1 1	11	11	0	1 1	51	33	18	1 1	28	117	-89	1 1	24	38	-14	1 1	15	23	-7	1 1	20	32	-12	0 1	36	51	-14	1 1	25	59	-33	0 1	67	31	36	Q-15 分かち入力の方がスムースに打鍵
1	1 1	18	68	-50	1 1	27	48	-20	1 1	41	27	14	1 1	41	47	-6	1 1	12	53	-40	1 1	31	23	7	1 1	25	52	-26	1 1	26	40	-14	1 1	41	53	-12	Q-43 きれいに削除された
2	1 1	22	67	-44	1 1	40	27	13	1 1	27	52	-25	1 1	-	-	-	1 1	-	-	-	1 1	35	40	-5	1 1	43	147	-103	1 1	22	138	-115	1 1	34	35	-1	Q-23 分かちの方がなめらかに打鍵
2	1 0	21	21	0	1 1	27	29	-1	1 1	27	49	-22	1 1	33	82	-48	1 1	-	-	-	1 1	43	26	16	1 1	27	58	-31	1 1	30	71	-41	1 1	30	41	-10	Q-24 作文打ちすることができない
2	1 0	27	44	-17	0 0	51	70	-18	1 0	21	34	-13	1 0	-	-	-	1 0	23	71	-47	1 0	36	47	-11	1 0	33	78	-45	1 0	30	48	-17	1 0	39	76	-36	Q-36 さらに検討を行ない
2	1 1	25	30	-4	1 1	54	72	-18	1 1	41	58	-17	1 1	27	64	-36	1 1	27	86	-59	1 1	63	53	9	1 1	40	58	-18	1 1	31	87	-55	1 1	35	46	-11	Q-38 文章内容を考える時間も
2	1 1	23	85	-62	1 1	29	78	-48	1 1	-	-	-	1 1	33	67	-33	1 1	-	-	-	1 1	38	35	3	1 1	28	73	-44	1 1	17	59	-42	1 1	27	59	-31	Q-47 打鍵についてグラフ
3	1 1	21	69	-47	1 1	85	69	15	1 0	37	60	-23	1 1	-	-	-	1 1	53	48	4	1 1	58	57	0	1 1	56	173	-116	1 1	40	58	-17	1 1	118	96	21	Q- 6 句読点で区切られた文
3	1 1	7	25	-17	1 1	25	70	-44	1 1	37	42	-5	1 1	81	38	42	1 1	35	50	-14	1 1	33	39	-5	1 1	28	57	-28	1 1	49	64	-15	1 1	39	72	-33	Q-19 同じテキストを打った場合
3	1 1	27	48	-20	1 1	-	-	-	1 1	23	42	-19	1 1	30	40	-9	1 1	36	65	-28	1 0	38	63	-25	1 1	36	67	-31	1 1	18	50	-32	1 1	37	70	-32	Q-45 同等の速度で打つ習慣
4	1 1	22	32	-9	1 1	36	54	-18	1 1	29	51	-21	1 1	50	55	-4	1 1	35	98	-63	1 1	-	-	-	1 1	34	95	-61	1 1	30	104	-73	1 1	92	60	31	Q- 2 本当にそうであろうかという疑問
4	1 1	176	65	110	0 1	240	90	149	1 1	88	79	9	1 1	81	87	-6	1 1	66	73	-6	1 1	76	57	19	1 1	96	74	22	1 1	52	102	-50	1 1	86	59	27	Q-21 速を計算した結果
4	1 1	23	15	7	1 1	18	62	-43	1 1	26	40	-14	1 1	27	48	-20	1 1	27	52	-24	1 1	81	27	54	1 1	30	63	-32	1 1	17	45	-27	1 1	31	45	-14	Q-35 極めて有効と思われる
4	1 0	25	84	-59	0 1	55	103	-48	1 1	23	42	-19	1 1	38	82	-43	1 1	29	53	-23	0 1	-	-	-	1 1	38	111	-72	1 1	25	57	-32	1 1	37	130	-93	Q-46 両手打鍵の高速なもの
4	0 1	143	70	73	1 1	79	94	-14	0 1	69	51	17	0 1	67	90	-23	1 1	83	62	20	1 1	75	86	-11	0 1	79	76	2	1 1	52	76	-24	1 1	132	112	19	Q-49 大切である
5	0 1	97	61	35	1 1	30	111	-80	0 1	-	-	-	0 1	94	66	27	0 1	53	82	-28	1 1	181	70	110	1 1	49	90	-41	1 0	69	89	-20	1 1	-	-	-	Q-13 入力ではかなり波打った図形
5	1 1	7	43	-36	1 1	28	68	-40	1 1	-	-	-	1 1	35	31	4	1 1	29	37	-7	1 1	-	-	-	1 1	30	57	-26	1 1	30	62	-32	1 1	42	72	-29	Q-17 １行当りの平均打鍵時間
5	1 1	27	36	-9	1 1	43	99	-56	1 1	29	51	-22	1 1	29	56	-27	1 1	26	65	-38	1 1	27	47	-19	1 1	31	102	-71	1 1	89	94	-5	1 1	47	62	-15	Q-34 作業効率のごく単純な目安
6	1 1	31	68	-36	1 1	42	81	-38	1 0	56	38	18	1 1	52	52	0	1 1	35	63	-27	1 1	51	66	-14	1 1	25	75	-49	0 0	42	70	-28	1 1	43	60	-17	Q- 9 そこで得られたローマ字テキスト
6	1 1	25	56	-31	1 1	18	69	-51	1 1	71	41	29	1 1	91	68	22	1 1	51	54	-2	1 1	35	50	-15	1 1	47	84	-36	1 0	32	61	-28	1 1	40	61	-21	Q-20 中央値どうしでの比較
7	1 1	28	66	-37	1 1	56	60	-4	1 1	-	-	-	1 1	47	60	-12	1 1	24	60	-35	1 1	111	67	44	1 1	31	58	-27	1 1	33	58	-24	1 1	32	66	-33	Q-39 取り扱うことにする
8	1 1	95	54	41	1 1	-	-	-	1 1	63	59	4	1 1	35	55	-20	1 1	54	135	-81	1 1	47	114	-67	1 1	37	108	-70	1 0	95	85	9	1 1	39	71	-31	Q-16 記憶させてから打鍵すれば
8	0 0	-	-	-	1 0	29	92	-63	0 0	120	51	69	0 0	153	79	74	0 0	-	-	-	0 0	172	97	74	0 0	92	96	-4	1 1	343	84	259	0 0	129	172	-43	Q-18 記憶打ちでは１行分表示してから
8	0 1	75	67	8	1 1	97	111	-13	0 1	139	65	73	0 1	104	106	-2	0 1	141	56	84	1 1	176	72	103	1 1	60	72	-11	0 1	73	69	4	0 1	121	70	50	Q-33 実際の応用例として
10	0 1	-	-	-	1 1	-	-	-	0 1	111	60	50	0 1	94	102	-8	0 1	133	128	4	0 1	103	113	-9	0 1	77	239	-162	0 1	82	75	6	0 1	-	-	-	Q- 7 画面上で推敲する
10	1 1	-	-	-	1 1	91	80	11	1 1	148	45	103	1 1	60	97	-37	1 1	65	51	14	1 1	54	56	-1	1 1	58	74	-15	0 1	68	57	10	0 1	78	65	13	Q-42 統計値によって例外打鍵を削除
11	1 0	-	-	-	0 0	70	68	1	1 1	-	-	-	1 0	-	-	-	1 0	-	-	-	1 0	-	-	-	1 0	48	89	-40	0 0	73	75	-1	0 0	69	95	-25	Q-40 英数字を削除する
13	0 0	52	69	-16	1 1	-	-	-	0 1	38	46	-8	1 0	105	57	48	0 1	75	135	-60	0 1	50	73	-23	1 1	36	80	-43	0 1	74	72	1	1 1	130	111	18	Q-26 事象サーバに転送
13	1 1	46	93	-46	1 1	60	84	-24	1 1	34	53	-19	1 1	55	100	-44	1 1	-	-	-	0 1	73	59	14	1 1	59	124	-65	1 1	46	79	-33	0 1	102	142	-39	Q-28 時間分解能１０ミリ秒
15	0 0	82	102	-19	0 0	184	57	126	0 1	72	61	11	0 1	-	-	-	0 0	85	68	17	0 0	246	76	169	0 0	188	81	106	0 0	122	69	53	0 0	236	57	178	Q- 4 書き変えて打鍵データ
16	1 0	74	132	-57	0 0	-	-	-	1 0	134	98	36	1 0	-	-	-	1 0	-	-	-	1 0	158	184	-26	0 0	136	145	-8	1 1	-	-	-	1 0	79	199	-119	Q-14 分かち入力の方がべた入力より
17	1 1	24	48	-24	1 1	349	50	299	1 1	-	-	-	1 1	121	55	66	1 1	82	76	6	1 1	36	98	-61	1 1	192	72	119	1 1	37	90	-53	1 1	329	77	251	Q- 3 表示させ
17	1 0	731	84	647	0 0	167	93	73	1 0	193	70	123	1 0	266	137	129	1 1	133	102	31	0 0	343	100	242	0 0	-	-	-	0 0	154	81	72	0 0	257	109	147	Q-25 考えながら文章を打鍵する
20	1 0	52	76	-24	0 0	110	77	32	1 0	322	70	252	1 0	87	62	24	1 0	30	98	-68	1 0	-	-	-	0 0	134	98	35	0 0	-	-	-	0 0	232	111	121	Q-12 全平均打鍵時間と比べて
21	1 0	89	55	34	1 0	251	73	178	1 0	43	49	-5	1 0	60	89	-29	1 0	67	88	-21	1 0	235	81	153	1 0	87	127	-40	0 0	-	-	-	0 0	99	108	-9	Q-48 高速に打鍵できることがわかった
22	1 0	-	-	-	1 0	287	108	179	1 0	219	88	130	1 0	86	119	-33	1 0	108	101	7	1 0	152	102	50	1 0	104	110	-6	0 0	192	160	31	0 0	185	92	93	Q- 5 記憶したら打ち始める
23	1 0	-	-	-	0 0	79	80	-1	1 0	284	63	221	1 0	136	119	16	1 0	-	-	-	1 0	100	115	-15	1 0	799	134	665	0 0	-	-	-	0 0	190	124	65	Q- 8 影響が出てしまう
23	0 0	-	-	-	1 0	43	89	-46	0 0	65	62	3	0 0	104	75	28	0 0	-	-	-	0 0	-	-	-	0 0	157	122	35	0 0	-	-	-	0 0	-	-	-	Q-22 作文打ちにもあてはまるか
25	0 0	-	-	-	1 0	180	106	74	0 0	178	60	118	1 0	81	129	-48	0 0	-	-	-	0 0	-	-	-	1 0	131	123	7	0 0	191	114	77	0 0	186	188	-1	Q-11 区切り前は母音の
26	0 0	-	-	-	0 0	86	120	-33	0 0	79	57	21	0 0	112	90	22	0 0	82	89	-6	0 0	-	-	-	0 0	80	107	-27	1 0	60	86	-26	0 0	-	-	-	Q-30 接続可能なキーボード
26	1 0	88	48	40	0 0	155	75	80	1 0	-	-	-	1 0	138	53	85	1 0	36	85	-48	0 0	85	71	13	0 0	102	62	39	0 0	167	81	86	0 0	71	68	2	Q-32 経時変化を示したものである
27	0 0	114	86	28	1 0	191	90	101	0 0	87	73	14	0 0	97	102	-5	0 0	167	58	108	0 0	98	81	16	0 0	87	81	5	0 0	90	63	27	0 0	107	76	30	Q-41 長い思考時間は入っていない
29	0 0	399	85	314	1 0	1068	134	934	0 0	620	65	555	0 0	147	87	60	0 0	201	52	149	0 0	491	88	402	0 0	619	63	556	0 0	-	-	-	0 0	171	152	19	Q-37 実際に論文を書くといった
38	0 0	-	-	-	0 0	214	102	111	0 0	219	56	162	0 0	100	117	-17	0 0	105	69	35	0 0	176	110	66	0 0	167	179	-12	0 0	361	75	286	0 0	137	123	14	Q-31 局所的運動動作を測定するため
41	0 0	231	86	144	0 0	-	-	-	0 0	340	61	278	0 0	370	102	268	0 0	172	121	51	0 0	96	94	2	0 0	137	74	62	0 0	123	74	48	0 0	159	72	86	Q-44 指が高速に動いている
48	0 0	-	-	-	0 0	-	-	-	0 0	109	64	44	0 0	101	96	5	0 0	173	71	101	0 0	120	77	43	0 0	101	98	2	0 0	128	73	54	0 0	145	94	50	Q-29 付随のキーボードを使い
52	0 0	109	98	11	0 0	182	120	62	0 0	189	60	129	0 0	86	120	-33	0 0	100	103	-2	0 0	121	96	25	0 0	-	-	-	0 0	243	138	105	0 0	298	89	209	Q-27 単位とほぼ一致していた
57	0 0	159	67	91	0 0	186	96	90	0 0	199	55	144	0 0	159	97	61	0 0	113	64	49	0 0	-	-	-	0 0	479	84	395	0 0	148	97	51	0 0	202	96	105	Q-10 大むね所要時間に差がない