

# TUTコード練習環境の提案

総合政策学部 4年 武田林太郎

## 概要

キーボードからの入力はコンピュータ利用の基本行為であるので、その入力方法を適切に選ぶことは重要である。現在の日本語入力の主流は、読み仮名から漢字に変換するかな漢字変換であるが、これに対してキーの入力によって漢字を直接入力する漢字直接入力という方法がある。漢字直接入力は、入力中の作業の思考が中断される漢字選択行為がない利点があるが、普及していないのが現状である。その大きな要因は、漢字直接入力の練習環境がないことである。

本提案は、漢字直接入力の一つである TUT-Code を普及させるためにその練習環境の構築方法に関するものである。筆者が TUT-Code を練習した経験を通じて感じた、有効な練習方法、練習の際に苦勞した点を挙げる。これらの反省を踏まえて、練習の手助けになるソフトウェアを展望として述べる。

## 目次

1	はじめに	1
2	今までの練習方法	1
2.1	TUT-Code Home Page の練習テキスト	1
2.1.1	ひらがな練習	2
2.1.2	漢字練習	2
2.1.3	テキストの改善	4
2.2	実用的に使用	4
3	作成した TUT-Code 練習環境	5
3.1	SFCxTYPE	5
3.1.1	概要	5
3.1.2	機能	5
3.1.3	練習方法の特徴	6
3.1.4	評価と展望	6
3.2	TUTSearch	8
3.2.1	概要	8
3.2.2	機能	8
3.2.3	展望	8
4	今後の展望	9
4.1	Web サイトの活用	9
4.2	検索漢字練習	10
4.3	TUT-IME の作成	11
4.4	かな漢字変換と漢字直接入力の比較	12

4.5	コンポジションタイピング練習環境 . . . . .	13
-----	----------------------------	----

# 1 はじめに

筆者はこの文書を TUT-Code によって書いている。TUT-Code は漢字直接入力のみならず初心者が容易に覚えられるようになっている。この文書を読んでいる多くの方は、かな漢字変換によって漢字を入力しているだろう。かな漢字変換とは、漢字の読みをひらがなで入力した後に漢字に変換する入力方法である。例えば「情報社会」を入力する場合、「じょうほうしゃかい」を入力して、スペースキーを押して変換させ、エンターキーを押して選択する。この際に入力するキーストローク数は(ローマ字入力の場合)、“jouhousyakai+SPACE+ENTER”の計14回である。「じょ」を”zyo”と打ったり、一度で変換できない場合はそれ以上打つことになる。

一方、漢字直接入力は「情報社会」を入力するのに”xyrrgldy”とキーを打てば入力できる。たったの8文字である。それ以下でもそれ以上でもない。ここでその仕組みは説明しないが、単純に漢字直接入力であれば少ないキーストローク数で漢字を入力できる。かな漢字変換よりも漢字直接入力の方が入力が速いだろうと想像できる。

TUT-Code は漢字直接入力の一つであり、初心者にも入りやすい特徴がある。筆者はこの文書を書く5か月前から TUT-Code の練習を始めた。以前はかな漢字変換によって漢字を入力していたのだから、TUT-Code に関しては初心者とも言える。TUT-Code を使えるようになっての利点は沢山ある。むしろ、TUT-Code によってかな漢字変換ができるのだから不便になることはないのである。

2節では、筆者が今まで TUT-Code をどのように練習してきたのか、その反省を含めて述べる。これによって、読者が TUT-Code を実用で使えるかどうかを判断できることを期待する。3節では、筆者が TUT-Code を練習するにあたって作成したソフトウェアを紹介している。4節では、今後 TUT-Code を練習する人にとって便利であろう練習環境を提案している。これらの練習環境は今後実装されるので、TUT-Code を利用したいと思った人は活用して欲しい。

ぜひ、多くの方が漢字直接入力によってより快適にコンピュータを利用してもらいたいと心から願っている。

## 2 今までの練習方法

ここでは、筆者が今までに TUT-Code を練習してきた過程を述べる。TUT-Code を本格的に練習するようになったのは2003年の6月中半からである。それまでは、多少さわってみた程度でかなも覚えていないと考えて良い。

まずは、TUT-Code Home Page<sup>1</sup>の練習テキストを使用して練習を始めた。それから4か月後の10月中半に増田式テキストを用いてさらに漢字コードを覚える練習をした。実際に TUT-Code を入力の主体にできるようになったのは最近になってからである。それまでは TUT-Code よりも、ローマ字入力でのかな漢字変換を使うことの方が多かった。

### 2.1 TUT-Code Home Page の練習テキスト

TUT-Code Home Page の練習テキストでは、ひらがなと、2ストローク漢字のうちの320字を修得した。

---

<sup>1</sup><http://www.crew.sfc.keio.ac.jp/~chk/>

### 2.1.1 ひらがな練習

ひらがなに関しては、「TUT-code かな入門（１）」、「TUT-code かな入門（２）」、「ひらがなスピードアップ（１）（２）」のテキストを練習した。タイピングのホームポジションや運指法に関する説明は、既に熟知していたので読み飛ばすことにし、かなキーコードだけに集中して覚えた。「TUT-code かな入門（１）」では、「あ行」か「な行」までのひらがなを覚えるが、ここでは一通り全てのかなキーコードの位置を覚えたいので、「TUT-code かな入門（２）」を練習してかなを網羅することにした。

次に「ひらがなスピードアップ（１）（２）」の練習文を練習した。ここでは、スピードアップ練習と言って徐々にタイプスピードを速くしていく練習方法を取るが、かな漢字変換による入力など他の入力方法で既にタイピングをしている人で、タイピングの姿勢を以前に習得している人であれば、このスピードアップ練習を忠実にこなさずに漢字の練習へと進んでかまわないであろう。

スピードアップ練習をしっかりと行わない理由としては、以前に他の入力方法でタイプしていた人にとって新しい入力方法によるタイプ速度向上の練習はもどかしいものであるからだ。スピードアップ練習をしてゆくうちに気が付くことでもあるが、やはり以前の入力方法で入力したくなる。実践で TUT-Code を使用することによってある程度まではタイプスピードは上がってゆく。

### 2.1.2 漢字練習

ひらがなのキーコードを覚えたら速度の向上を考えずに、漢字直接入力の醍醐味であるところの漢字コードを練習する方が楽しく練習できる。そこで「漢字練習テキスト A 1」を練習した。筆者の漢字練習記録は次のようになる。

漢字テキストを練習するために全体的に気を付けることは以下の２点である。

- あまり漢字コードを覚えよると身がまえずに、何度も打つことで体に覚えさせる気持ちで練習する
- 覚えられないと思った漢字もあとで復習するので気にしない

テキストのすすめ方として 1-20 課までは、1 課に対して 12-20 分程度時間をかけ、1 課 1 課しっかりと練習する。21 課からは、1 課に対して 8-18 分程度時間をかけ、1 度の練習に 5 課程度練習すること。これを基本に練習してゆけばよい。

また、1 課に対しては次のような手順で練習すればよい。

1. その課で覚える漢字を 1 つ 1 つ、それぞれ 5 回ほど入力する
2. 漢字コードを見ないようにしながら、5 回ずつ入力する。忘れたら確認してすぐにまた見ないようにする。
3. 全ての漢字を 1 通り見ないで入力する。忘れたら確認してすぐにまた見ないようにし、1 つでも忘れた場合は再度覚える漢字の始めから入力する。
4. 漢字が入ったテキスト練習に入り、まずは全てのテキストを打つ。このときに漢字コードを忘れたら見てよい。
5. もう一度テキストを打つが、その課の漢字コードを一度も見ずに全てのテキストを入力できるまで練習する。

表 1: 漢字練習記録

練習日	練習課	練習時間	覚えた総漢字数
2003/06/28	1-3	40	+18
2003/06/29	4-10	90	18+30
2003/08/05	2-7	30	48+0
2003/08/06	8-12	30	48+12
2003/08/08	12-17	60	60+30
2003/08/08	13-20	60	90+6
2003/08/14	21-22	30	96+14
2003/08/14	21-25	50	110+14
2003/08/14	25-31	60	124+35
2003/08/14	31-32	20	159+14
2003/08/16	26-30	30	159+0
2003/08/17	31-35	40	173+7
2003/08/17	36-40	50	180+28
2003/08/18	31-40	50	208+0
2003/08/18	41-42	15	208+14
2003/08/18	41-48	60	222+35
2003/08/19	41-48	40	257+0
2003/08/19	49-50	10	257+7
2003/08/19	1-50	45	264+0
2003/08/19	51-55	40	264+28
2003/08/20	51-55	20	292+0
2003/08/21	56-60	50	292+28
2003/08/31	全て	65	320+0
2003/09/02	31-40	30	320+0
2003/09/05	31-50	45	320+0
2003/10/08	1-50	40	320+0

6. 再度, その課の漢字だけを全て漢字コードを見ずに入力して, 忘れていなかったら次の課に移る.

早く練習したいと思って, 自分が覚えていると思われる漢字やひらがな等を飛ばしてテキストを練習しないこと. このテキストでは, 漢字を覚えること以外に, よくある文章を打つ「パターン」を覚えることもでき, 漢字だけを練習してはその「パターン」を体に覚えさせることができないからである. 「パターン」の例は, 「です.」「あった.」といった読点を含んだ一連の入力パターンや, 「情報」, 「議論」といった, 良く使う熟語に対する入力パターンなどである.

筆者の経験から, 各課に対してかける時間は徐々に少なくなる. 漢字を覚えるスピードは速くなってゆくように感じられる. 初めは, 練習がもどかしいと思われるかもしれないが, 20 課あたりまで取りあえず努力して欲しい.

表 1 からも分かるように集中して漢字 320 字を練習したのは 1 か月である. 練習方法をさらに洗練させることによってさらに短い期間で主要漢字を覚えられるような, 練習カリキュラムを作成する予定である.

### 2.1.3 テキストの改善

この練習テキストの問題点は次の 4 点である.

- 初めの課で練習する数字は実用で使用する頻度は低い
- 既に覚えた漢字がひらがなになって後の課に出ている
- テキトベースでの練習環境しかない
- 練習する漢字 320 字よりも一般的に日常で使用する漢字が他にある
- 320 字以外のテキストが (オンライン上に) ない

この問題点に対しては後に説明するソフトウェア, SFCxTYPE<sup>2</sup>によって解決したい.

## 2.2 実用的に使用

TUT-Code Home Page の練習テキストをひとつおりに練習したあとは, 実用的に TUT-Code によって入力するようになった. ここで, 新たに漢字を覚えたいときには TUT-Code 漢字検索 [図 2] することによって, 1 つずつ覚えることとなった.

実用的に使用することで問題になることは, 以前の入力方法にたよってしまうことである. レポートやメールなど, 急遽文章を書く必要がある場合には, やはり覚えたての入力方法で入力するのは遅いと感じる. しかし, TUT-Code を使う頻度を上げてゆくことによって徐々にスピードは速くなる. 一方, 入力する頻度が下がる以前の入力方法のタイピングスピードは下がっていった. これによって, TUT-Code と以前の入力スピードの差がちぢまることによって, 数か月後には TUT-Code による入力が主流になった.

現在は, TUT-Code の入力をさらに上げるための漢字練習ができるソフトウェアを完成させたいと考えている.

---

<sup>2</sup>sfctype.tex



図 1: SFCxTYPE の概観

## 3 作成した TUT-Code 練習環境

ここでは、筆者が今までに作成した TUT-Code を練習するためのソフトウェアについて説明する。

### 3.1 SFCxTYPE

#### 3.1.1 概要

SFCxTYPE は、あらゆるタイピング方式にも対応したタイピングソフトウェアを目標としている。現在は、かな入力と TUT-Code に対応しているが、練習テキストに漢字が含まれているので、かな入力での練習は実質できないことになる。

SFCxTYPE の開発背景は、学部 3 年後期の研究会テーマを選ぶ際にタイピングソフトウェアを作成しなかったことから始まる。当時は TUT-Code を練習したいという希望よりも、研究会テーマを決めることに重きが置かれて SFC-TYPE が開発されたことになる。期間は約 5 か月、人数は 3 人といった環境での開発であり、実現された機能はそれほど多くない。

研究会後に共同開発が終わってからは、共同開発をしていた 1 人と開発を続けることを決め、SourceForge.jp<sup>3</sup>に登録して続行する形となった。現在、筆者が利用している SFCxTYPE は、テキストだけを TUT-Code Home Page<sup>4</sup>の練習テキストに移植している。これによって、Web を閲覧する事なく TUT-Code の練習ができるようになっている。

#### 3.1.2 機能

SFCxTYPE の主な機能は以下の 4 つである。

- 入力方式が変えられる
- 練習テキストを変えられる

<sup>3</sup><http://sourceforge.jp>

<sup>4</sup><http://www.crew.sfc.keio.ac.jp/~chk/>

- 練習成果をサーバに送信する仕様になっている
- 入力コードを補助できる

1つめの機能である入力方式を変えるためには、その入力方式のコードを定義したファイルを用意する必要がある。そのファイルを SFCxTYPE をインストールしたディレクトリの data ディレクトリの下に置き、strokeCodeList.txt に追加したいファイル名と入力方式の名前を書き込むだけで、新しい入力方式によって、タイプ練習ができるようになる。

練習テキストを変えるためには、入力方式を変える方法と同様に、練習テキストを用意して data ディレクトリの下に置き、stageList.txt に追加したりファイル名と練習テキストの名前を書き込むだけで、新しいテキストを練習できるようになる。

SFCxTYPE によって練習された結果は、サーバへ送信される仕組みになっているので、分析するツールを作成するだけで、自分の練習状況を管理することができる。現在の仕様としては、タイプしたキーと、次のタイプへの遷移時間を保持している。これを分析することによって、練習者のタイピングの特徴などを推定できると考える。

例えば TUT-Code を練習している場合には、漢字コードを忘れてしまうことはしばしば起こるが、練習テキストに相当する入力コードを表示する補助機能が付いている。これによって、初心者でも容易に新しく入力コードを練習することができる。

### 3.1.3 練習方法の特徴

SFCxTYPE の練習方法は、SFC のタイピング教材として利用されている「TUT タッチタイピング」の練習方法を参考にした。それを踏まえた主な特徴は以下の3つである。

- 入力されたキーを出力しない
- テキストと違った文字を入力しても続けて入力できる
- 練習に関するアドバイスを表示できる

1つ目の特徴は「TUT タッチタイピング」の特徴でもあり、入力されたキーを出力しないことによって、間違いに臆病にならない、間違いに対しての処理を習慣化することを防ぐ、出力を気にせずに練習できるといった効果が期待できる。初心者がタイピングを練習する際に気を付けることの一つとして、タイピングミスの悪い習慣を付けないことが大切である。ある特定の文字を入力するとき、よく間違えるパターンができてしまうと、間違えることも体が覚えてしまい、間違えてから再入力するまでの動きが一連の動作として身に付いてしまう。初心者の練習の際にこれを防ぐためにも、入力されたキーを出力させないことは効果的である。

2つ目の特徴も一つ目と同様であり、間違いに対してあまり気にしないように配慮している。タイピング練習の際には一連の練習を終えてから、振り返って間違いを確認させている。これも、「TUT タッチタイピング」の練習方法を参考にして部分である。

3つ目の特徴に関しては、例えば TUT-Code Home Page の練習テキストのように、それぞれの練習の際に色々と気を付けることがあると想定して、アドバイスが表示できるようにした。

### 3.1.4 評価と展望

SFCxTYPE の今後の改善点は主に7つ挙げられる。

- 入力方式と練習テキストの追加を容易にする
- テキストの階層化ができるようにしたい
- TUT-Code に特化したテキストを用意したい
- 書ける漢字は入力できる
- 誤字判定アルゴリズムを洗練する
- サーバの仕組みを作る
- ソフトウェア事体の動作を速くする

「入力方式と練習テキストの追加を容易にする」に関してだが、現在新しい入力方式や練習テキストを追加する場合には、練習者がインストールディレクトリ以下のファイルを編集する必要がある。コンピュータの利用に慣れている人ならば容易にできるが、タイピング練習者がそうであるとは考えにくい。そこで、ある程度の入力方式や練習テキストを事前にサポートしていることはもちろんだが、SFCxTYPE のインターフェースで追加の操作ができることが望ましい。さらに、入力方式や練習テキストは Web で共有できるようにして、テキスト作成者と練習者を分離させるモデルが将来的に実現できると良い。このモデルが実現されると、練習者はいつでも最新の練習テキストを楽しめる。

練習テキストの追加を行うと、練習の順番を階層構造にしたいと思うことがしばしばある。TUT-Code Home Page の練習テキストもまず、「かな」と「漢字」の大項目に分かれ、かな練習には「TUT-code かな入門（１）」、「TUT-code かな入門（２）」、「ひらがなスピードアップ（１）」、「ひらがなスピードアップ（２）」に分かれている。SFCxTYPE はテキストの構造化ができる仕様になっていないので、練習テキストを追加できる仕様のまま開発を進めるのであれば、いずれテキストの構造化ができるようにするかを吟味する必要がある。

テキストの構造化をするかどうかの議論に関わってくるのだが、SFCxTYPE を TUT-Code を練習するのに特化したいという展望がある。人を幸せにするソフトウェア開発の重要点であるところの、対象をしぼることに当たる。

「書ける漢字は入力できる」について説明すると、現在の SFCxTYPE はテキストに準じた文字しか判定できない。例えば、「日本での八年間はたのしかった。」というテキストを入力する際に「たのしかった」を「楽しかった」と入力すると間違いになる。「たのしかった」でも「楽しかった」でも正解になるようにすれば、覚えた漢字をさらに強く記憶させることになる。

「サーバの仕組みを作る」に関しては実装予定ではあったので早く実現させた方が良い。後にも述べるが、タイピング練習者たちが切磋琢磨し合って練習できるように、練習成果をインターネットで共有する仕組みを実現したい。

「誤字判定アルゴリズムを洗練する」に関してだが、SFCxTYPE の誤字判定アルゴリズムは、先行文献を参考にしたものではない。誤字判定アルゴリズムを研究しながら SFCxTYPE に反映させていきたい。

最後に、SFCxTYPE の実行速度が遅いので改良の余地に入れたい。



図 2: TUTSearch の概観

## 3.2 TUTSearch

### 3.2.1 概要

TUTSearchは、TUT-Codeの文字コードを検索するソフトウェアである。かなや漢字からTUT-Codeに相当するキーを検索したり、一連のキーストロークから相当するかなや漢字を逆に検索できる。例えば、「山田太郎」を入力するためにTUT-Codeでどのように打てば良いのか分からない時に、TUTSearchを使って検索することができる。検索することで「山田太郎」は「hr ks tr mf」とキーを打てば入力できることが分かる。ここで検索された「hr ks tr mf」はQwerty配列に準じたキーに相当する。

TUTSearchの開発背景は、TUT-Codeを実用で使う程度に入力できるようになってから、文書を作成している時などにTUT-Codeをすぐに検索できると便利であったからである。TUTSearchはTUT-Codeの基本的な漢字を覚えた後に、頻繁に使用されると予想される。

TUTSearchは、<http://web.sfc.keio.ac.jp/~s00581rt/tut/tutsearch/><sup>5</sup>からダウンロード可能である。Java Web Startを利用しているので、どのプラットフォームでも容易にインストールして使用できる。

### 3.2.2 機能

TUTCodeの機能は、検索と検索履歴の保存の2つである。

検索機能は概要でも述べたように、かなや漢字からTUT-Codeに相当するキーを検索したり、一連のキーストロークから相当するかなや漢字を逆に検索する機能である。TUT-Codeにない「炒」などの漢字は検索されずに「null」を出力し、逆検索でTUT-Codeに相当しないキーストロークを入力した時には見つからない旨を伝える。

検索履歴の保存機能は、検索した漢字と日付をファイルに保存する機能である。TUTSearchを使って検索を行えば行うほどファイルにその履歴が蓄積され、そのファイルを分析すれば検索した漢字の頻度や、どの時期にどのような漢字を検索していたか、漢字の忘れる間隔などが分かるようになっている。

### 3.2.3 展望

TUTSearchの改善したいものは以下の3点である。

- 検索結果の表示方法を増やす
- 保存機能を強化する

<sup>5</sup><http://web.sfc.keio.ac.jp/~s00581rt/tut/tutsearch/>

- IME と連動できるようにする

1つ目の「検索結果の表示方法を増やす」に関しては、現在 Qwerty 配列でのキーストロークでしか検索結果を表示できないが、Dvorak 配列など他のキー配列でも表示したい。また、コンピュータを初めて利用する人が TUT-Code によってタイピングを習得するとなれば、Qwerty 配列によるキーの位置を見付けるだけでも困難である。そこで、キー配列による表示だけでなく、キーボード画面を表示することによる検索結果の出力をすべきである。

2つ目の「保存機能を強化する」に関しては、Ctrl+S(コントロールキーを押しながら'S'を押す)を明示的に押しながら検索、もしくは検索した後に Ctrl+S を押すことで検索履歴の保存ができるのだが、保存するタイミングを利用者が明示することは面倒ではないか、ということである。

検索履歴を明示的に保存する必要があったのは、保存したくない漢字を検索することが多いからである。例えば、「調」という漢字を検索する時に、「調べる」と入力して検索した場合に、「調」「べ」「る」の3つの言葉それぞれに対して検索が行われる。この時にもし検索履歴を保存してしまうと、「調」だけでなく「べ」と「る」に対しても検索履歴が残ってしまう。

検索履歴を残す目的は、良く調べる漢字だけを履歴から分析して一括して練習したいためである。忘れやすくかつ頻繁に使用する漢字を割り出す方法を考えつつ、現在のように明示的に検索履歴を残すかどうか、保存のタイミングを吟味する必要がある。

また、Java Web Start による TUTSearch は、Java のセキュリティ仕様のためにファイルの書き込みが制限されていて検索履歴の保存ができない。これは、署名を付けることによって解決することが分かっているので早急に対処する。

3つ目の「IME と連動できるようにする」に関しては、かな漢字変換をした時に次回から TUT-Code でも入力できるように、変換した漢字に相当するキーストロークを表示する仕組みにしたい。これは後に説明する TUT-IME[4 節]につながることである。

また、TUT-Code には登録されていない漢字がいくつか存在し、それらの漢字を検索した場合には、TUT-Code として未登録のキーストロークの流れに、検索漢字を割り合てる機能を予定している。これには、TUT-Code を入力するための IME と連動しなければならない。

## 4 今後の展望

筆者の TUT-Code の練習の経験を踏まえて必要だと思われる TUT-Code の練習環境を提案する。

以下に説明する練習環境はどれも漢字直接入力の普及を目指すものであり、多くのプラットフォームに対応したいと考えている。少なくとも Windows, Mac OS, Linux といった現在利用されている主流のプラットフォーム上で動作するように設計したい。スタンドアロンのソフトウェアであれば Java Web Start を利用し、インストールを容易にさせたい。また、Web アプリケーションによる実装であれば、ブラウザを利用するほとんどの人が使い慣れたインターフェースによっての利用が可能となる。

### 4.1 Web サイトの活用

まず初めに、TUT-Code の本家のサイト<sup>6</sup>が更新されていないことが、TUT-Code を普及させるための弊害となっている。2004年1年現在の状況を見ると、最新の更新が99年3年から止まった

---

<sup>6</sup><http://www.crew.sfc.keio.ac.jp/~chk/>

ままである。TUT-Code を利用するためのプラットフォームも Window98 や NT といったものに関して紹介されている。

TUT-Code 本家のサイト管理者と話し合い、筆者が今後本家のサイトを更新することとなった。これによって、TUT-Code に関する情報を更新することで多くの利用者が、容易に TUT-Code を使用して漢字直接入力に親しめるようになるだろう。

サイトの内容だが、実際には次のように変更する予定である。

- Windows XP, Mac OS X 等, 新しいプラットフォームでのインストール方法を載せる
- 練習ソフトウェアを載せる
- TUT-Code 利用者が互いにコミュニケーションできる場を設ける

## 4.2 検索漢字練習

TUT-Code の漢字コードを検索して覚えるうちに、何度も同じ漢字を検索していることに気が付く。これは筆者の TUT-Code の練習経験から言えるのだが、英単語など何かを覚えようとしたことがある人ならこのような経験はだれにでもあることは容易に想像できるであろう。暗記するために 1-2 回程度試行しただけでは短期記憶にしかならず、それを長期記憶させるには何度も繰返す必要がある。

何度も再確認してしまうような漢字に対しては、ある期間に集中させて練習してそれを長期記憶にさせたい。TUTSearch[図 2] は漢字コードを検索するだけではなく、その検索履歴を保存していることは前にも述べたが、この TUTSearch の履歴を利用することによって、多く検索する漢字を集中して練習することが可能である。教材になっているような漢字を覚えるためのテキストは、一括し漢字を覚えられるメリットがあるが、やはり自分が頻繁に使用する漢字に焦点を当てて練習すると、高い意欲で取り組める。そして日常的にその漢字を使用することでさらに長期記憶につながってゆく。

検索漢字練習のソフトウェアは開発中であり、その中で 2 つ問題が出てきた。問題の 1 つは、検索履歴のファイルを読み込んだときに、検索した文字から本当に知りたかった文字を判断しにくいことである。

筆者の TUTSearch による履歴を例に取ってみると、2003 年 8 月 26 日に「漢字」という文字を検索しているが、その 3 か月後の 2003 年 11 月 20 日にも同じく「漢字」という文字を検索している。この場合、8 月の時点では「漢字」の「字」という文字を検索したくて「漢字」と入力し、11 月の時点では「漢」の文字を検索したくて「漢字」と入力したと考えられる。もう 1 つの考えとしては、単に 3 か月後であったので「漢」という文字を忘れてしまった、もしくは「字」という文字を忘れてしまったとも捉えられる。

検索対象の文字から本当に検索したかった文字を断定することは不可能であるが、ある程度検索したかった文字を推定することは可能である。推定の方法は、検索対象の文字から覚えた文字を除外することである。覚えていない文字しか検索しないのは明らかなことである。しかしこのようにして検索したかった文字を推定するためには、覚えた文字が何であるかを予め分かってなければならぬ。この「覚えた文字」を特定しにくいことが次の問題につながる。

ところで、知りたい文字コードを検索する時に、明示することがもう 1 つの解決策である。先ほどの例であれば、8 月の時点で「字」を検索したかったのなら「漢字」と始めに入力し、その後

「漢」を削除してから検索すればよいのである。明示的に検索する機能は TUTSearch に既に実装され、Ctrl+S で初めて検索履歴としてファイルに保存される。

明示的に検索する操作は面倒で、検索履歴の量が少なくなる可能性が高いことがこの欠点である。筆者の 5 か月間の検索履歴は高々 235 字である。Ctrl+S を押して明示的に検索するのは不便であり、実際に検索した回数はこの 2 倍以上になるであろう。

2 つ目の問題であるのが、文字を覚えたことをどのようにして判断するかである。覚えたという感覚はあいまいなもので、自分自身で覚えたと認識したものであっても忘れてしまうことがある。「覚えた」という感覚をレベルに分けることによって、このあいまいさは解決される。漢字を覚えるのであれば、ある 1 つの文字に対する練習数、打ち間違った失敗数、再び練習した間隔の 3 つの値を使って計算することによって点数化してレベル分けを試してみれば良い。

この計算はそれほど容易ではなく、計算方法も吟味する必要がある。また、記憶に関する先行研究を調べることが必要である。計算が難しいのは、例えば「漢」の文字を 1 か月間隔で 12 回練習し、最後の 3 回からずっと間違えていない場合と、「字」の文字を 4 週間間隔で 12 回練習し、最後の 6 回からずっと間違えていない場合ではどの文字がより覚えているかを考えると良い。3 つの値が色々変わるにより、最後に練習した日からどれだけ経過しているのかによっても覚えた度合いは変わる。

### 4.3 TUT-IME の作成

現在、いくつかの TUT-Code を入力する入力装置 (Input Method) は存在するが、どの IM もそれぞれ使いやすさが異なっている。筆者が利用してきたいくつかの TUT-Code の IM の良かった点を挙げると、

- インストールが容易である (1 フォルダ内で完結する)
- TUT-Code のテーブルがテキストファイルで定義されているので変更できる
- 1 つキーを押すとそこから始まる漢字のリストを表示する補助機能がある
- かな漢字変換ができる
- かな漢字変換の辞書登録ができる
- まぜ書き変換ができる
- 1 ストローク目の入力文字を出力してくれる

また、悪かった点は、

- (TUT-Code 入力環境にするための) インストールが不便である
- スペースを含む文字コードとかな漢字変換が競合する
- かな漢字変換の辞書能力が低い
- 数字を半角にしたり、句読点をピリオドにするようなカスタマイズができない
- 1 ストローク目の入力文字を出力しないために、現在の入力の順番が分からない

- ソフトウェアに依存する、挙動が異なる

のようなものがあり、どの IM も良い点と悪い点を共にそなえていた。

また、これらの入力装置は TUT-Code を入力することも専門としていない。TUT-Code を利用するためには、いくらかのカスタマイズをする必要がある。ソフトウェアをインストールする際に、標準の設定から変更するような経験がない人にはこれは困難なことである。TUT-Code はかなに対する文字コードに規則性を持たせることで、初心者にとって分かりやすい。しかし、その TUT-Code を利用するまでの環境設定をすることが難しくては意味がない。

TUT-Code を多くの人に利用できるようにするためには、TUT-Code に特化した入力装置を作成して無料で配布すべきである。また、主要のプラットフォーム上で利用できなければならない。複数のプラットフォームを利用している人にとっては、各プラットフォームによって入力方法が多少でも異なっては入力と思うようにできないのである。

例えば、筆者が利用している Mac OS X 上で動作する TUT-Code の入力装置では、かな漢字変換を併用させる場合、Shift キーを押して始めの文字を入力する。一方、Windows XP で利用している入力装置では、漢字変換したい文字を予め入力した後にスペースキーによって変換を行う。このように、プラットフォームに共通でない IM を利用することは非常に不便である。

開発する IM には、かな漢字変換をした場合にその漢字に相当する漢字コードを表示する機能を付加する。検索といった明示的に漢字を検索することは、文章を作成している最中にはとても不便に感じることもある。ある文章を作成している時には、同じ漢字を何度も利用することが多いが、1 度検索してもすぐに忘れてしまうことが良くあり、何度も検索してしまうことはしばしばある。かな漢字変換時に同時に漢字コードを検索し表示することでこの手間がはぶける。この機能を実装するときには、表示のタイミングや記憶したと判断された時には表示しないなど考える必要がある。

#### 4.4 かな漢字変換と漢字直接入力の比較

漢字直接入力がかな漢字変換よりも早く入力できることが証明できれば、漢字直接入力の普及につながる。漢字直接入力は理論上かな漢字変換に比べてキーストロークが少なくすむことは言えるが、実際に入力が早いかを一般の人に実証する必要がある。かな漢字変換の方がキーストローク数が多いのは、少なくとも漢字を変換する際にスペースを 1 回押し、漢字を決定するための Enter キーを 1 回押すことになり、最低でも 2 回のキーストロークが必要になるからである。

キーストローク数だけでタイピング速度の測定ができない理由は、かな漢字変換と漢字直接入力では、漢字を打つ際の思考プロセスが違うと考えられるからである。かな漢字変換は、漢字を「音」に変換し、その音の読み方を入力する。その後に変換して選択という過程を経る。一方、漢字直接入力の場合は、漢字を想像し、その漢字に当てはまる漢字コードを思い出して入力する。いずれの入力過程も、タイピング練習を重ねることによってそのプロセスを飛ばすことができる。打とうと思った文字のイメージから、そのまま体が覚えていて打つといった、先程の思考過程を一度に飛ばして入力できるようになる。ここまで入力できるようになれば、ストローク数が少ない漢字直接入力の方が速く入力できると考えられるが、一般の人々がタイピング練習をしてここまで打てるようになるには相当な時間を要する。

一般の人がタイピング練習をする程度で、かな漢字変換と漢字直接入力の速度の違いがどれほどかを測るには、実際に集計して比べてみるのが分かりやすい。集計するデータの収集は、多くの練習者の成果を Web 上で共有することで実現できる。タイピング練習の環境を提供している Web サービスはいくつかあり、その参加数の多さから Web 上でのタイピングによるデータ収集は妥当

である。

Web上タイピングの目的は、かな漢字変換と漢字直接入力の入力速度の比較であるから、入力方法を問わずに利用してもらおう。この場合、タイピング練習の事前にどの入力方法で漢字を入力するかを明記してもらい練習を始めてもらう。

また、ユーザの練習状況の履歴を取ることで、かな漢字変換と漢字直接入力の上達速度の違いも測定する。このように、入力方法を問わないWeb上での練習環境を提供することによって、入力方法同士の違いが見えてくることを期待する。

オンラインにする頻度が比較的に少ないユーザに対しては、Java Web Start を用いたスタンドアロンの練習環境を提供したい。練習の履歴はオンラインになった時にサーバ側へ送信するような仕組みにする。

#### 4.5 コンポジションタイピング練習環境

現在市販のタイピングソフトウェアはほぼ全てコピータイピングの練習ソフトウェアである。コピータイピングとは、あらかじめ打つべきものが決められている文字を打つことである。コピータイピングの練習が主流になっている背景に、タイピング速度を期待されていたのがタイプライタであったことが挙げられる。

現在、一般の家庭にコンピュータが普及するようになってからは、一般の人々がタイピング練習をする必要がでてきた。一般の利用者はコピータイピングよりもコンポジションタイピングであることが多い。コンポジションタイピングとは、自分で考えたものも打つことである。コンポジションタイピングの例としては、メールを書く、レポートを書く、チャットをする、検索を行うなどと、コンピュータを現在利用する一般ユーザのほとんどん行為である。また、プログラミングなどもコンポジションタイピングであるので、コンピュータで仕事をする人の大半はコンポジションタイピングをしていることになる。

コピータイピングよりもコンポジションタイピング練習の需要が高くあるべきだが、コンポジションタイピングを主としている一般の人々は、市販のコピータイピングでタイピング速度を上げていることで満足している場合が現状である。市販のソフトウェアもタイピングソフトウェアを称して、その本質であるところの練習の要素を欠いているものも見られる。

ここで、コンポジションタイピングを練習するためのソフトウェアを作成したいと考える。コンポジションタイピングの提案はもともと大岩研究室の方々から出ていたものであり、例えば写真やイラストイメージを表示し、それに相当する名前を入力するイメージタイプ練習がある。リンゴやバナナのイラストを表示してもらい、「リンゴ」や「バナナ」と入力することで正確にタイプできたとする。

他には音を聞くことによる調音タイピングや、チャットによる練習、クイズ形式で答える練習、課題を出しそれを制限時間内に解く練習などがある。特に最後の課題に取り組むタイピング練習などは、プログラミング言語の修得に応用できる。

コンポジションタイピング練習環境を構築する際に問題となるものは誤字判定になる。誤字判定の研究はコピータイピングでも様々な先行研究が残されているように、その特定が容易ではない。今後コンポジションタイピング練習環境を構築するにあたって、誤字判定をどのように行うかを吟味する必要がある。

誤字判定に関しては、厳密な誤字判定アルゴリズムを考える代わりに、それを人間に任せてしまうことを考えている。コンポジションタイピングの練習であるから言えることなのだが、例えばイメージタイピングで「リンゴ」を表示した場合に「林檎」「りんご」「Apple」「赤い果実」「筆者の

大好物」のいずれをタイプしても正解とする。赤くて丸いりんごのようなイメージを表示して、そこから想像するものは千差万別である。無数存在する正解の作成方法は、Webでのタイピング練習を想定しているので、練習者各々で正解のリストを作成、追加してもらえれば良いと思う。