3Q－5

キャラクタ推論による コマンド検索機能付きシェルの試作

桑畑文子　高橋正司　丸本 悟　道城謙治　今村二康　金岡泰保　富田真吾
（中国日本電気ソフトウェア株式会社）　（山口大学工学部）

１．はじめに

対話型システムの利用効率を下げる要因の一つに、ユーザによるコマンドの誤入力がある。ユーザは、誤入力を行なう度に正しいコマンドを入力し直さなければならず、特に、長いコマンド名や紛らわしいコマンド名では、手間がかかる。更に記憶があいまいであれば、複数回の試行が必要となる。

誤入力は、１～２文字の過不足あるいはタイプミスが主な原因と考えられ、この誤入力したコマンドを自動的に修正してくれる機能があれば、より効率的に対話型システムを利用できる。

筆者らは 、かねてより あいまい検索の一手法として研究中の「キャラクタ推論」[1][2][3] がこのコマンド修正の問題に適すると考え「キャラクタ推論によるコマンド検索機能付きシェル」[4]を試作した。

対話式のコマンド体系を持つシステムとしては、ｕｎｉｘのコマンドインタプリタの一つである「ｃｓｈ」を選んだ。

２．キャラクタ推論

２．１．キャラクタ推論とは

従来のキーワード検索は、「完全一致」、「部分一致」、「ワイルドカード」などによるパターン検索であり、キーとなる文字列と一致するパターンがない場合、そのキー文字列のパターンが存在しないという情報以外には何も得られなかった。

「キャラクタ推論」は、検索の母集団となる文字列群を持ち、その母集団内の全文字列と、キーとなる文字列（以後、「キー文字列」と呼ぶ）との類似度を計算する。そして、類似度の高いものを検索結果とする検索方法である。

「キャラクタ推論」は、キー文字列と一致するデータが無い場合に類似度の高いデータを候補として示すことによって、従来のパターン検索では扱えなかった部分をカバーできる。

「類似度」は、与えられた２つの文字列の共通する文字の状態や長さなどを基に計算される。その値は、０以上１００以下の整数をとり、２文字列間に共通文字が全くない場合は０、２文字列が完全に一致する場合は１００となる。

例えば、母集団を「実在するコマンド名」、キー文字列を「誤入力されたコマンド」として「キャラクタ推論」を実行すると、誤入力した文字がさほど多くない限り、ユーザの意図したコマンドが、類似度の高いものとしてあらわれてくることになる。

２．２．類似度

「類似度」は、比較すべき２文字列に対して、その文字列間の「共通文字数」、「連続共通文字数」、「共通する文字列のずれ具合い」、「キー文字列に対する共通文字の割合」などを要因として計算される。

「ずれ具合い」に関しては「先頭か最後が揃っていること」、他の要因に関しては「多いこと、大きいこと」が、それぞれ類似度を高くする因子となる。

類似度計算の概要を図Ｉに示す。図中、類似度(ruijido)は、推論の母集団となる データ内の １コマンド名(s) とその長さ(sl)、キー文字列(key) とその長さ(keyl)とを入力とし求められる。また、「WS()」は、文字列間の連続共通文字に関する重みを計算する関数であり、「WC()」は、２文字列間の共通文字に関する重みを計算する関数である。

```
ruijido=0
s'=s ,key'=key
(while:s',key'間に共通な文字がある)
  共通文字数が最大となるような位置を見つける
  先頭か最後が揃っていれば fit=1
        そうでなければ fit=0
  その位置における以下の要因を調べる
    • 共通文字の数        (com)
    • 連続共通文字列の数   (j)
    • 連続共通文字数の長さ(chain[j])
  共通であった文字をスペースに置き換え、
    新しい s', key' とする
  (for:i=1 to j)
    連続共通文字の重み(wchain)の計算
      wchain+=WS(chain[i],com)*chain[i]/keyl
  共通文字の重み(wcommon)の計算
    wcommon+=WC(fit,com,keyl)*wchain
  類似度(ruijido)の計算
    ruijido=wcommon*α/(|sl-keyl|/keyl+α)*100
```

<図Ｉ：類似度計算の概要>

３．キャラクタ推論付きシェルの概要

「コマンド検索機能付きシェル」は、以下の特徴を持つ。

①「キャラクタ推論」には、検索の母集団として「実在するコマンド名」のテーブルが必要である。「実在するコマンド名」は、「組み込みコマンド名」と「実行可能ファイル名」から成り、前者はシェルのデータとしてコマンド名のテーブルを持つ。後者については、シェル起動時に、新たにテーブルを作成する。

②コマンド未発見時のみ、「キャラクタ推論」を起動し、類似度の上位５件を検索結果として表示する。

③コマンド未発見時に「キャラクタ推論」を起動するか否かは、独自のシェル変数(conjecture)の有無に依存することとする。

４．結果

検索の母集団には、５８７件のコマンド名（平均５．７文字／コマンド）を用い、ヒット率と推論時間の計測を行なった。

４．１．ヒット率

前述の母集団の中から任意にコマンドを選び、誤修正を加える。これをキー文字列として推論を行なった結果において、誤修正前のコマンドが候補の１位となる割合（ヒット率）を計測した。誤修正は、以下の４種類とし、それぞれ１０００件の計測データを得た。

①不足：任意の１文字を削除する誤修正
②過多：任意の位置に任意のアルファベットを挿入する誤修正
③置換：任意の１文字を任意のアルファベットと置き換える誤修正
④転置：隣接する任意の文字を入れ換える誤修正

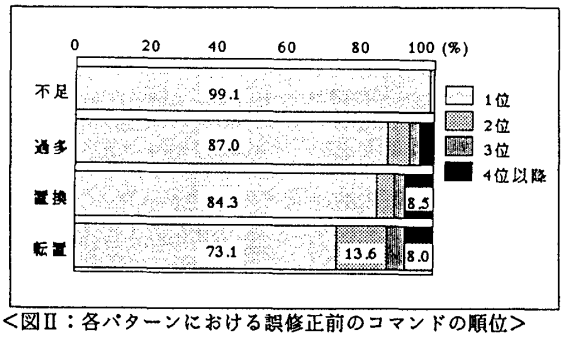
表Ｉ、図IIに、推論の結果における誤修正前のコマンドの順位の割合をパターン別に示す。

Prototyping of the Shell with command search function by the String Inference Module

Fumiko Kuwabata,Masashi Takahasi,
Satoshi Marumoto,kenji Dohjo,Tsuguyasu Imamura
NEC Software Chugoku,Ltd.
Taiho Kanaoka,Shingo Tomita
Yamaguchi University

<表Ⅰ：各パターンにおける誤修正前のコマンドの順位>

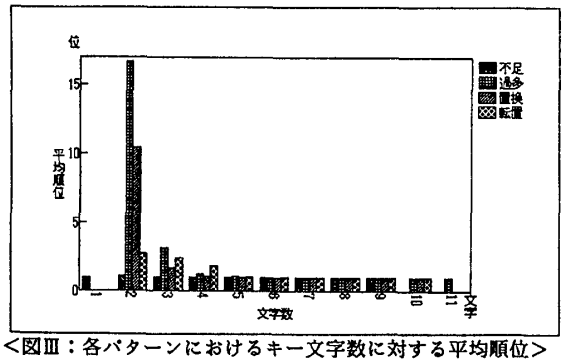
パターン	1位	2位	3位	4位以降（%）
① 不足	99.1	0.7	0.1	0.1
② 過多	87.0	6.3	2.4	4.3
③ 置換	84.3	4.9	2.3	8.5
④ 転置	73.1	13.6	5.3	8.0
平均	85.9	6.4	2.5	5.2



<図Ⅱ：各パターンにおける誤修正前のコマンドの順位>

ランダムな誤修正を加えたキー文字列の平均して８６％は、修正前のコマンド名を候補の１位としている。最高の値は「不足」における９９％、最低は「転置」における７３％である。パターンによってヒット率に違いがあるが、１位から３位までを加えると、どのパターンにおいても９０％を超えている。

図Ⅲは、上記の結果をキー文字列の文字数別にしたものである。縦軸には、その文字数のキー文字列における誤修正前のコマンド名の順位の平均をとった。



<図Ⅲ：各パターンにおけるキー文字数に対する平均順位>

２～３文字の比較的少ない文字数での平均順位は高くなっているが、それ以外の部分では、ほぼ１位となっている。

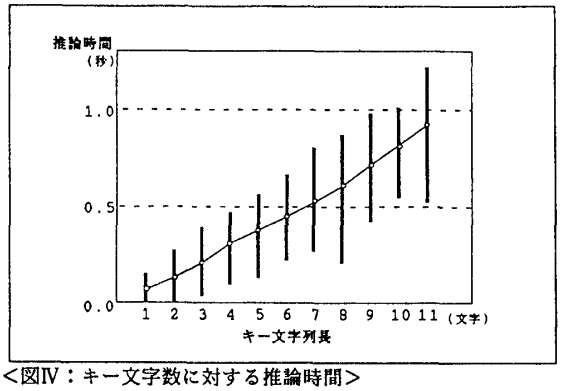
パターン別でみた場合、「不足」は文字数に関係なく、常に１位に近い。「過多」は、最小文字の２文字では、平均１７．６位という順位となっているが、３文字では、３．２位、４文字では、１．３位と１位に近づく。「置換」も同様である。「転置」では、２文字で２．８位と低いものの、３文字で２．５位、４文字で１．９位と収束が遅い。

４．２．推論時間

推論時間の計測では、任意のアルファベットの組合せによる文字列をキー文字列とし、１文字から１１文字までのキー文字列について計測した。

キー文字数と推論時間との関係を、図Ⅳに示す。

各文字数において、ある程度の開きはみられるが、１～１１文字程度では、文字数に対して推論時間はほぼ線形に増加している。



<図Ⅳ：キー文字数に対する推論時間>

５．考察

ヒット率において、修正前のコマンド名が候補の１位となる確率は転置パターンが最低であるが、１位から３位までの合計では置換パターンが最低である。これらの例において４位以降となったケースは、２文字から３文字程度の比較的短いキー文字列であった場合に偏っている。これは、各パターンにおけるコマンドの誤修正を全くのランダムに行なったため、１文字の置換や転置により、修正前のコマンド名より似ているコマンド名ができてしまう可能性が大きかったことを示す。

図Ⅲにおける「過多」の平均順位が、２文字（１文字のコマンド名に対し、ランダムなアルファベットを１文字加えた２文字のキー文字列での推論）で高くなっていることも、上記の理由による。

推論時間は、キー文字列と共通の文字を含む、母集団内のコマンド数に影響される。また、共通文字の並び方によっても推論時間は変わってくる。最悪の場合、推論時間はキー文字列の長さの二乗に比例するが、今回の結果では、線形に近いものになっている。また、１０文字以上のコマンド名は希であることから、コマンド検索への適用は、十分現実的であると考える。

６．今後の検討事項

今回の評価において、キャラクタ推論でコマンド検索を行なう場合のいくつかの問題点が明らかになった。

まず、シェルの誤入力で比較的起こり易い「転置」のヒット率を上げる必要がある。しかし、先述の通り、「不足」以外の誤修正パターンでは、修正前のコマンドよりも他のコマンドにより似ているキー文字列が出来てしまう可能性が大きいという問題がある。２つの文字列の文字の状態のみを入力とする現在のキャラクタ推論ではこの問題に対処できず、ヒット率が１００％になることは難しい。実用化する上でも問題となるため、他の要因として、コマンドの使用頻度、キーボード上のキーの位置、発音類似アルファベットなどを追加することを考えている。

次に、推論時間の問題である。線形増加の傾向にあるとはいえ、キー文字列が長い場合には、１ｓｅｃ以上の推論時間を要する。推論の高速化、特に比較的長いキー文字列で推論を行なう場合の高速化が必要であろう。

また、インタフェース面からは、現在は誤入力の後の支援だけであるが、コマンド入力途中における支援も実現したい。

７．文献

1) 森永、小林他：”データ検索のための推論法とＯＡシステムへの応用”，信学技報，OS87-13(1987).
2) 小林、森永他：”データ検索のためのキャラクタ推論について”，電気四学会中国支部連合大会(1987).
3) 久保田、高橋他：”あいまい検索を用いたファイル情報検索システムの試作”，電気四学会中国支部連合大会(1988).
4) 久保田、高橋他：”キャラクタ推論によるコマンド検索機能付きシェルの試作”，電気四学会中国支部連合大会(1989).