日本語入力における平仮名文字列の予測

1M－4

蓮井洋志　西野順二　小高知宏　小倉久和
福井大学情報工学科

1. はじめに

既入力文字列から次入力の予測を行う入力補助システムとしてRK[1]やDM[2]などがある。RKは過去の入力履歴からユーザが次に入力するつもりである文字列を予測し、ユーザに提示し、それをユーザが選択することでアルファベットの入力補助を行う。

本報告では、RKの予測アルゴリズムを改良して日本語入力において文節の入力を補助する方法を提案する。この入力補助インタフェースはRKの予測アルゴリズムを利用して、次入力の平仮名文字列の予測を行う。なお、この入力補助機能は仮名漢字変換システムWnnを改良して実現した。

2. 平仮名文字列の構造と文節の予測

RKは過去の入力履歴から次入力の文字列を予測するシステムである。過去の入力において頻出した文字列が予測結果になるために文章中でよく使われる文節や常套句の入力補助に向いている。しかし、RKは英語の予測システムで日本語入力には不適応な部分も多い。日本語入力に適したRKを実現するために、平仮名文字列の構造の特徴についてまとめる。

日本語入力においては文節単位の予測を行わなければいけない。日本語入力は普通仮名漢字変換フロントエンドプロセッサを用いて行う。仮名漢字変換フロントエンドプロセッサは文節毎に変換を行うために、平仮名表記は文節の区切り目で終わらなくてはいけない。文節の区切り目で終わっていない予測文字列は削除などの不要なキー入力が必要になる。

また、2文節以上の予測を行うと予測の的中率が悪くなる。日本語の文節間の文字の結合力は、文節内の文字の結合力と比較して弱い。そのために日本語入力における平仮名文字列の予測は、1文節毎に予測できることが望ましい。

3. 予測アルゴリズムとその実現

平仮名文字列の構造の特徴を考慮して、1文節毎の予測をするためにRKの予測アルゴリズムに改良を加えた。

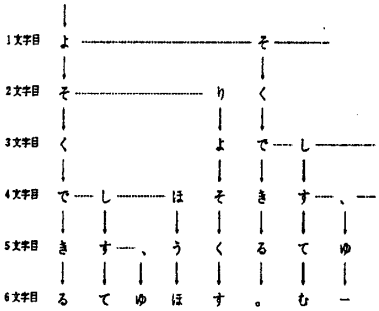

図1: 6-gramのtext modelの例

過去に入力された文字列のn-gramを作成する。検索の都合上、n-gramを木構造にし、各々のノードに文字と文字に関する情報と隣接したノードへのポインタを保存する。文字に関する情報には、n-gramの出現回数やその文字が文節末であるかどうかのフラグなどがある。過去の入力文字列が仮名漢字変換の結果、形態素解析によって文節毎にわかれているために、文節末であるかどうかがわかる。このn-gramの木のことをtext modelという。また、n-gramの長さは(日本語の文節の長さ+1)文字を想定して6文字とした。図1にtext modelの例を示す。

text modelはuum.2を起動した時に今までの入力履歴ファイル内の文字列をtext modelに登録する。変換結果を確定した時に確定した文字列の平仮名表記を入力履歴ファイルに書き込む。ユーザが予測を指令した(予測キーを入力した)時に新たに入力履歴ファイルに追加された文字列をtext modelに登録する。text modelではn-gramの出現回数によって各々のノードは順位付けされる。回数の大きいものから照合の優先順位が高くなる。ノードの照合順位の変更はtext modelが更新された時に行う。

ユーザが予測キーを入力するとバッファ内の文字列をもとに予測し、第1候補の予測文字列をバッファの中に出力する。次候補キーを入力すると前の予測候補を削除し、第2候補以降の予測文字列を出力する。予測候補は10個用意する。

バッファ内に“かなかんじ”があるときに予測キーが押された場合を考える。予測1文字目は10種類の異

Prediction method to aid typing Japanese
Hiroshi Hasui,Jyunji Nisino,Tomohiro Odaka,
Hisakazu Ogura in Fukui University

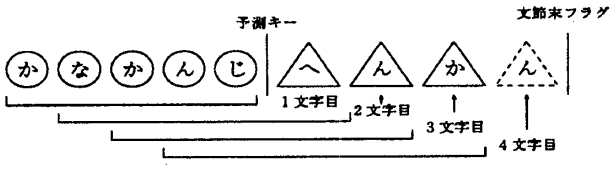

図 2: “かなかんじ” の予測

なった文字を生成する。“かなかんじ” と text model 内の 6-gram を照合する。照合の結果、“かなかんじへ” と一致する。一致した 5-gram の後の ‘へ’ が第 1 候補の予測 1 文字目になる。第 2 候補の予測 1 文字目は ‘へ’ の同レベルポインタの指し示しているノードの文字になる。この例では ‘」’ になる。これは “かなかんじ」” という 6-gram と “かなかんじ” が一致したことを表している。同レベルのノードがない場合は、1 文字頭を詰めた “なかんじ” と 6-gram を照合して同様に決める。第 10 候補が決まるまで照合する。

予測 2 文字目以降の生成は、10 種類の予測 1 文字目各々に対して行う。先程の例で予測 1 文字目が ‘へ’ の場合、“かなかんじ” を 1 文字シフトさせた “なかんじへ” と 6-gram の照合を行う。“なかんじへん” が一致し、‘ん’ が予測 2 文字目になる。3 文字目は 1 文字シフトさせた “かんじへん” と 6-gram を照合して同様に決める。4 文字目の予測で ‘ん’ が予測される。‘ん’ は文節末フラグが True なのでそこで予測を打ち切る。他の予測 1 文字目についても同様に行う。

本報告の予測機能は Wnn のクライアントである uum を改良して実現した。改良した仮名漢字変換システムの構成図を図 3に示す。

4. システムの動作解析と予測の効果

入力補助の効果を推定するために本論文を入力、推敲し、予測機能の動作を解析した。本論文に頻出する長い文節である「仮名漢字変換システム」、「平仮名文字列」に対して予測機能を適用した。予測結果を以下に示す。ただし、“()” 内は、予測に使用されたバッファ内の文字列で、“|” のあとは予測文字列をしめす。下の行の | の後の文字列は次候補の予測文字を示す。

(ひら) | がな | もじ | れつ
(かな) | いことが
| らず
| かんじ | へんかん | ふろんと
| していく

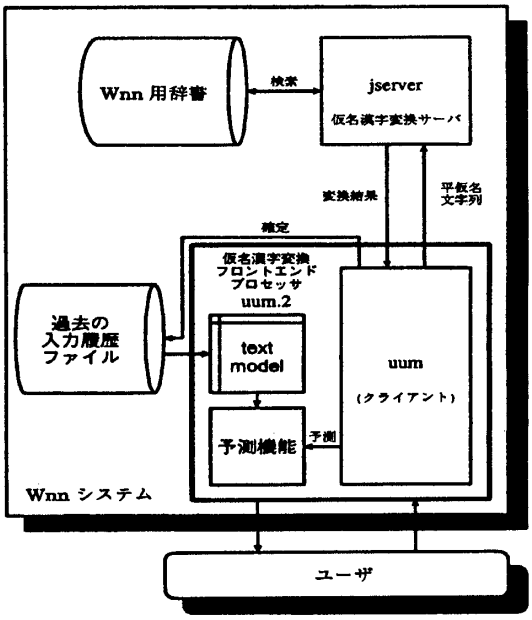

図 3: 改良した Wnn システムの構成図

(かなかんじへんかんし) | ていく
| すてむ

予測機能はあらゆる文字列に対して適用が可能である。しかし、入力の手間を省くという観点からすれば、長い文節の予測には有効であるが、短い文節に対しては効果が薄い。

また、「よそくされた。」の入力を意図して「よそく」をもとに予測したが、「よそくされる。」が候補として現れるために「よそくされた。」が候補の中に入らなかった。予測文字列の予測第 1 文字目はすべて異なった文字にしているためにこのような問題が生じる。予測の失敗には活用のある語が絡んでいる場合が多く、活用が複雑な付属語の入力補助には向かない。

課題としては、頻出する活用の複雑な単語を含む文節に対応するために、予測 1 文字目のノードの出現回数が大きい場合には 1 文字目が同じで 2 文字目以降が異なった予測文字列を生成するようにアルゴリズムを改良することなどがある。

参考文献

[1] John J.Darragh,Ian H.Witten,and Mark L.James: “The Reactive Keyboard:A Predictive Typing Aid ”, COMPUTER, Nov. 1990, pp.41-49

[2] 増井 俊之, 太和田 誠: “操作の繰り返しによるマクロの自動生成”, ヒューマンインタフェース, May 1993, pp.65-71