4P−12

# 仮名漢字変換における最尤候補選択アルゴリズムの実験

酒井貴子，下村秀樹，並木美太郎，中川正樹，高橋延匡
(東京農工大学　工学部　電子情報工学科)

## 1．はじめに

仮名漢字変換は今やワードプロセッサの入力手段としても定着し，学習情報や単語の意味，文法属性などを利用して変換を行い，変換精度を向上させる工夫が一般的になされてきている[1]．我々もこれを重視し，変換結果を決定する評価尺度を一つに限定せず，どんな情報でも自由に利用して変換が行える仮名漢字変換システム（OS/omicron 仮名漢字変換システム第２版）を設計し[2][3]，今回そのプロトタイプを作成した．本稿では，これを用いて行った本仮名漢字変換方式の実現性の評価実験について述べる．

## 2．本仮名漢字変換における結果選択アルゴリズム

本仮名漢字変換システムでは，変換結果を決定するのに評価尺度を限定しないという思想から，考えられるすべての候補に対して結果としてのもっともらしさの評価を行い，評価最大の候補を変換結果とする方式を採用した．ここでいうもっともらしさとは，単語の意味や文法属性による接続の善し悪しや前に使用されたという情報（学習）などによって判断され，評価値として表現される．例えば，「学習された単語は評価値＝2，他は評価値＝1」という評価基準を設定した場合，“変換”が学習されていれば，“高精度な返還”より“高精度な変換”の方が評価値が１点高くなり，結果として優先する仮名漢字変換を実現できる．このように，評価値を設定する評価系を変更するだけであらゆる情報を変換に利用することができる．

しかし，日本語の曖昧性を考えると，１回の変換でユーザの望みの結果を返す評価系を実現することは非常に難しい．そこで，再変換回数が平均して少なく済むような仮名漢字変換が現実的なアプローチであると考えた．以上のことから我々は，**入力文字列に対して考えられるすべての変換結果を作り，もっともらしさの評価の高かった順番に結果を返す方式（最尤候補選択アルゴリズム）**を設計し，本システムで実現することにした．

## 3．最尤候補選択アルゴリズムの実現方法と評価実験

最尤候補選択アルゴリズムを実現すると，すべての変換結果を生成することから，変換結果が膨大になることが予想される．例えば，入力に対して変換結果が 100 個生成され，もっともらしさの評価の結果，ユーザの欲しい変換結果が 100 番目と評価されてしまったとする．すると，ユーザは再変換を 100 回も行わなくては欲しい結果が得られないことになる（図1）．

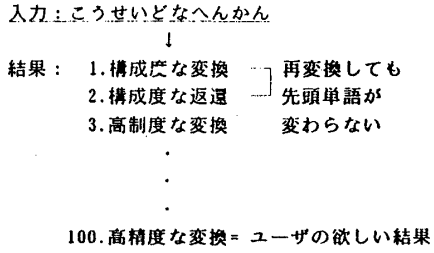


図1 最尤候補選択アルゴリズムの問題点

また図1 のように，評価によっては再変換しても先頭単語が変化しない可能性があり，ユーザインタフェースが悪い．そこで，変換結果を一つ一つ別々に評価する形ではなく，単語をノード，単語間の接続をパスとする木構造に表現して，この問題を回避することにした．各単語は評価の最も高かった単語を子供として持ち，子供は評価値の高い順番にソートされるので，この木をたどっていけば評価値の高い順番に結果を取り出せる（図2）．これによって，ユーザインタフェースに対応しようと考えた．

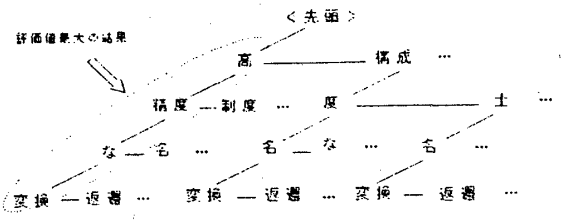


図2 “こうせいどなへんかん” の変換結果を木構造に表現した例

また，メモリ容量や変換に要する時間が実現できるオーダーになるかどうかは実際に調べてみる必要がある．そこで，今回作成した本システムのプロトタイプを用いて，この評価実験を行った．実験内容，実験方法，及び結果を次に示す．

---

Experiments on an algorithm for the selection of the best candidate in Kana-Kanji Translation

Takako SAKAI, Hideki SHIMOMURA, Mitarou NAMIKI, Masaki NAKAGAWA, Nobumasa TAKAHASHI
Department of Computer Science, Faculty of Technology,
Tokyo University of Agriculture and Technology

実験内容

(1)入力に対して生成される変換結果の木の大きさ（ノード数）を測定し，必要なメモリ容量を推定する．

(2)現段階における変換時間を測定し，実現性を検証する．

実験方法

本システムのプロトタイプで仮名漢字変換を行い，実験内容の(1)，(2)を測定した．入力データは，技術系の論文２本を１文節，句読点までの長さに分割した２種類のベンチマークテキスト[4]からランダムに 100 データ選び，それらを仮名に直したものを使用した．測定は，考えられるすべての変換結果を作成する場合と文法を用いて枝刈りを行う場合の２種類について行った．枝刈りに使用した文法は，人間が 200 文程度の文章を解析して，一度でも出現した品詞並びを文法接続可能と定義したもので，研究室内で作成された[5]．本実験を行う前に，入力用の 100 データに対してこの文法を用いて接続可能な結果だけを生成するように，文法表のカスタマイズを行ってから実験を開始した．

実験結果

(1)入力文字数に対して作成された変換結果の木のノード最大数を図3 に示す．

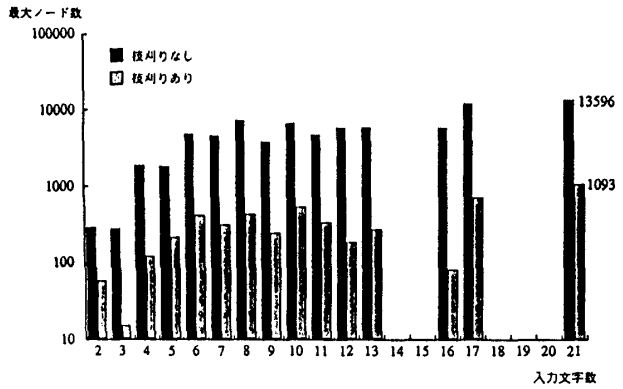


図３ 入力文字数に対する最大ノード数

ノードは木のリンクの情報や単語の表記，品詞情報などを持ち，メモリ容量／1ノードは 30 byte であるから，今回測定したデータでは最大で約 400K byte，枝刈りを行うと約 33K byte という見積りになる．今回使用したベンチマークテキストは１文節の平均が 6 文字，句読点までの長さの平均が 23 文字であり，これらのメモリ容量の平均はそれぞれ次のようになった．

１文節　　：約　42K ( 5K) byte

句読点までの長さ：約 200K (20K) byte

（注）括弧内は枝刈りを行った場合の平均容量

したがって，文法接続が不可のものについて枝刈りを行えば，変換に必要とされるメモリは，１入力につき数 10K byte のオーダーで収まることがわかった．また，今回の文法を用いて枝刈りを行うと，すべての結果を作成したときの平均 5.6% まで変換結果を限定できることがわかった．今回使用した文法表についてはまだカスタマイズの余地があり，今後も枝刈りの効果については測定を続けたい．

(2)プロトタイプ（実行環境は MC68020,20MHz）での１単語あたりの平均単語検索時間は 3.3[ms] で，木作成時間は 0.3[ms]（枝刈りを行うと 3.1[ms]），評価時間は 3.1[ms]（枝刈りを行うと 1.1[ms]）であった．具体例として，１文節，句読点までの長さに対する平均変換所要時間を表1 に示す．

表1 １文節，句読点までの変換に要する平均時間

| | １文節 | 句読点 |
|---|---|---|
| 平均ノード数 | 1334 (87) | 6309 (303) |
| 平均単語検索時間[s] | 0.3 | 2.5 |
| 平均木作成時間 [s] | 0.4 (0.2) | 1.6 (2.0) |
| 平均評価時間 [s] | 2.6 (0.07) | 38.6 (0.5) |
| 所要時間の合計 [s] | 3.3 (0.57) | 42.7 (5.0) |

（注）括弧内の数値は枝刈りを行った場合の所要時間

文法に従って枝刈りを行った場合，現在は１文節は平均 0.57[s]，句読点までの長さは平均 5.0[s]程度の時間を変換に必要とする．プロトタイプでは単語検索，木の作成，評価をそれぞれ独立に行っているため，現段階では無駄な単語検索も多い．こうしたことを改良し，今後は処理時間を短縮できるように最適化をはかる必要があるだろう．

## ４．おわりに

我々は，仮名漢字変換における最尤候補選択アルゴリズムを設計し，そのプロトタイプを実現し，変換に必要とされるメモリや時間の見積りを行った．その結果，文法による変換結果の限定効果が数値的に明らかになり，本アルゴリズムが実用化できるオーダーで実現できることが計測できた．今後は変換時間を短縮するように最適化すると共に，変換結果のもっともらしさの評価に対する実験を行い，本アルゴリズムの変換精度に及ぼす効果を測定して行きたいと考えている．

## 参考文献

[1]酒井他：“日本語ワードプロセッサにおける仮名漢字変換の変換処理の探針テスト”，情報処理学会第42回全国大会5Q-4，1991

[2]下村他：“OS/omicron 仮名漢字変換システム第２版の設計思想”，情報処理学会第42回全国大会5Q-1，1991

[3]本宮他：“OS/omicron 仮名漢字変換システム第２版の設計と実現”，情報処理学会第42回全国大会5Q-2，1991

[4]本宮他：“日本語ワードプロセッサの仮名漢字変換の解析と評価”，情報処理学会研究会報告 90-HI-33，1990

[5]下村他：“形態素解析を利用した日本語スペルチェッカ”，第31回プログラミングシンポジウム報告集，情報処理学会，1990