階層化概念辞書の高速検索アルゴリズム

7E－2

小山雅史　伊与田敦　青江順一

徳島大学工学部

1．はじめに

概念相互が上位／下位・部分全体関係で結ばれた知識の体系（以後階層化概念と呼ぶ．）はかな漢字変換における複合語処理，同音語の判定，同型語の読み分け，更には機械翻訳システムの意味処理等に広く利用されている(2)(3)(4)．そしてこれらの分野においては，階層の判定回数は非常に多く要求されるので，階層判定の高速化は重要な課題である．

本稿では，まず同音語判定における共起概念のマッチングについて考察を行い，その上で，概念パターンの高速検索手法を提案する．また実験により，本手法の有効性を示すと共に理論面からの評価も行う．最後に，今後の課題及び本手法の応用性について触れる．

2．階層化概念における検索

同音語の判定においては，各判定語それぞれの共起関係となる概念とのマッチングにより判定を行う．共起概念が複数存在する場合，概念間の階層関係が明確でなければ，概念の数だけ検索を繰り返さなくてはならず，非常に効率が悪い．

解決策として，仮に全体の階層化概念を持ち，その各概念から共起関係にある判定語へのリンクを持たせる場合を考慮しても，概念階層の大きさとコンピューターのメモリ容量を比較すると全く現実的ではない．

従って本稿では，各判定語の共起概念リストを階層化する手法を提案する．提案する構造（図１参照）は木構造で考えると，各判定語の階層化概念は全体の階層化概念の部分木になっている．

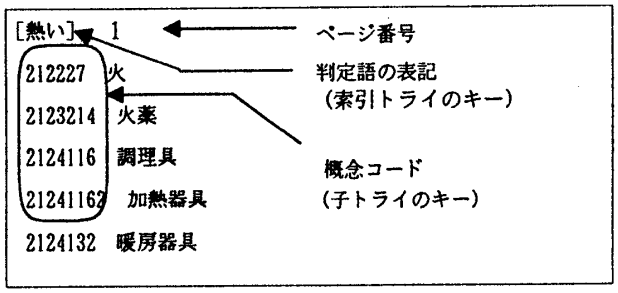

図１判定語の概念リスト例

本手法では階層化概念を効率的に実現するために，トライを用いる(5)．トライに与えるキーは概念間を繋ぐノードにそれぞれ番号を割り振った記号列（以後概念コード（図１参照）と呼ぶ．）で，これ自体が階層構造内の各概念のアドレスになっている．これによりトライの特長の入力概念数に左右されない高速な検索が可能になる．メモリ面からも各判定語の階層概念の１つ１つはさほど大きなものではないので，トライを用いて構築しても問題はない．

しかしこの場合には，作成されたそれぞれのトライの管理面が問題になる．従って次節でこの問題を解決するデータ構造とそれを構築及び検索するアルゴリズムを示す．

3．データ構造とアルゴリズム

3．1．データ構造

本技法で用いたデータ構造を以下に示す．

（1）判定語それぞれの概念コード（図１参照）をキーとするトライ．このトライは辞書に格納される．以後これを子トライと呼ぶ．

（2）（1）とは別に全判定語の表記（図１参照）をキーとするトライ．このトライは主記憶上に置かれる．以後これを索引トライと呼ぶ．

An Effcient Algorithm for Retrieving
Hierarchical Concept
Masafumi Koyama, Atsushi Iyota, Junichi Aoe
Department of Information Science
and Intelligent Systems,
University of Tokushima
2-1 Minamijosanjima, Tokushima, Tokushima 770 Japan

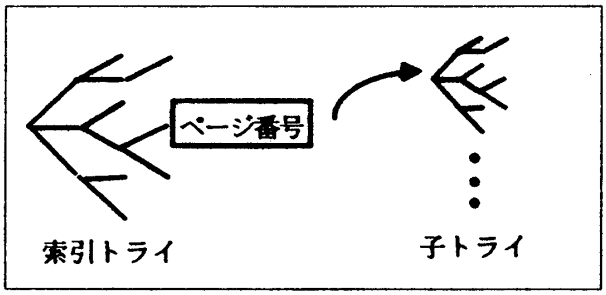

図２データ構造

３．２．アルゴリズム

３．２．１　構築アルゴリズム

３．１で示したデータ構造を構築するアルゴリズムを以下に示す．

（１）全判定語の索引トライを構築する．この際，各判定語に固有のページ番号（連番）を与えトライの末尾ノードに格納する．

（２）各判定語について子トライを構築し，（１）のページ番号を基に辞書内のページ番号の位置に格納する．

結果として図２に示す様な，トライを２重にした構造になる．

３．２．２．検索アルゴリズム

ここで本手法で用いた検索アルゴリズムを以下に示す．前提として概念コードと判定語を検索キーとして与えた場合の概念マッチングとする．

（１）索引トライで判定語の表記を検索する．成功した場合はページ番号を得て（２）へ，失敗した場合エラー．

（２）辞書内のページ番号のページから子トライを読み込む．

（３）子トライで概念コードを検索し，検索結果を返す．

４．評価実験

概念数７００，平均コード長５の概念階層辞書[1]を用いて本手法と索引トライのみを用いた場合の１回当たりの平均マッチング時間の比率をグラフにしたものを図３に示す．表中の概念数は判定語中の関係概念数を表す．

この結果より，本手法はトライを用いない場合との比率はほぼ概念数に比例している事が分かる．これは理論的にもトライと線形探索の計算量の比率O（1）：O（N）に適っている．理想値

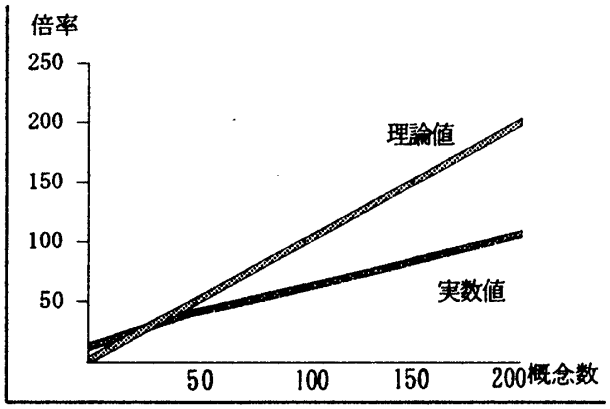

図３実験結果

（N倍）には及ばないが，これはトライの容量が大きくなると，その読み込み時間が無視できなくなってくるからである．この読み込み時間はトライの容量に比例するが，これはトライを細かいブロックに切り分けて格納する手法で改善できる問題である．現実には概念数が極端に大きくなる事はないであろうし，２，３０概念程度でも充分効果的であると考える．

５．まとめ

本稿では階層化概念におけるマッチングを高速に行う手法として，判定語各自に概念コードのトライを構築し，それを索引トライにより検索する手法を提案した．また実験によりその有効性と理論的妥当性を確認した．

今後は本手法を自然言語処理における同音語の判別に利用する予定であるが，その際の子トライ数の増加に伴いトライの効率化及び挿入・削除等の動的構成が必要である．また現段階では未使用である子トライの末尾部分に表層格情報等を格納する事で，より効率的な同音語判別が可能になると思われる．

参考文献

（１）ＥＤＲ電子化辞書，（株）日本電子化辞書研究所

（２）大島，阿部，湯浦，武市：”格文法による仮名漢字変換の多義解消”，情処学論，２７，７，pp.679-687（昭61-07）．

（３）宮崎，大山：”階層的単語属性を用いた同型語の自動読み分け方”，信学論（Ｄ），Ｊ６８－Ｄ，３，pp.392-399（昭60-03）．

（４）高松，西田：”動詞パターンと格構造に基づく英日機械翻訳”，信学論（Ｄ），Ｊ６４－Ｄ，９，pp.815-822（昭56-09）．

（５）青江：”キー検索技法－トライ法とその応用”，情報処理学会誌，Ｖｏｌ．３４，Ｎｏ．２，pp.244-251（1993）．