5B-9

形態素抽出アルゴリズムの高速処理方式

中村 修　田中 明通　菊池 英夫
ＮＴＴ情報通信処理研究所

１．まえがき

日本語解析処理の高速化を狙いとし，先に，木構造インデックスを用いた形態素抽出アルゴリズム，並列化ハードウェア構成法を提案し，ソフトウェアシミュレーションによる評価から性能向上の予測結果を示した[1]～[4]．

本稿では，実際の入力例文（かな列），辞書等のデータ，および上記構成のハードウェアを用いて行った評価実験から得られた，アルゴリズムの高速性ならびにハードウェアの並列化効果の検証結果を示す．

２．処理方式の概要

本処理方式では以下に示す手法および構成によって形態素候補抽出の高速化を図っている．

2.1 高速化手法

(1) インデックスの並列木構造化（図１）

・単語読みから作成した木構造インデックスの探索によって形態素候補を抽出

・上記インデックスを上下２階層に分割し，高頻度アクセスかつ少量の上位階層を並列化
低頻度アクセスかつ大容量の下位階層については，アクセス競合調停機能を付加して共有

(2) 抽出アルゴリズム実行の並列化（図２）

・入力かな列中の各かなを先頭位置として共有する全ての形態素の抽出処理は各々独立に実行可能であることから，形態素抽出開始位置を交互に並列動作可能な処理ユニット(PU)へ割付け

・入力かな列ならびに上位階層インデックスを全ＰＵへ割付け，形態素抽出処理を並列化

2.2 ハードウェア構成

並列化により１桁高速化を実現するため，上記アルゴリズムに適合した図３に示す並列処理構成で形態素抽出処理を試みる．並列度については，２０前後まで検討する必要があるが，ここでは４並列までの評価から並列化効果の傾向を把握することとした．

・１～４並列で形態素抽出を実行する処理ユニット（PUi）および共有メモリ（CMEM）から構成

・４個のＰＵからのＣＭＥＭへのアクセスが競合する場合には，これをＣＭＥＭ内で調停

・ホスト計算機に対しては，全ＰＵ内のメモリならびにＣＭＥＭが１つの連続した空間に見えるメモリインタフェースで接続

３．評価条件

(1) 評価モデル

・比較対象：かな漢字変換処理で一般的なテーブル形式のインデックス(主記憶常駐)を用いたソフトウェアを１ＰＵと同性能のＣＰＵで実行する方式

・抽出処理：１文に相当する入力かな列から，抽出可能な全形態素を抽出

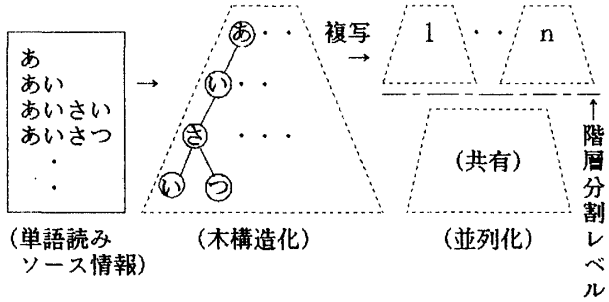

図１　インデックスの並列木構造化

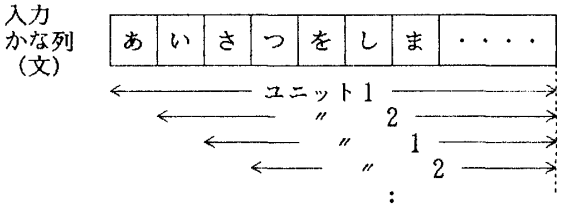

図２　抽出アルゴリズムの並列展開（並列度＝２）

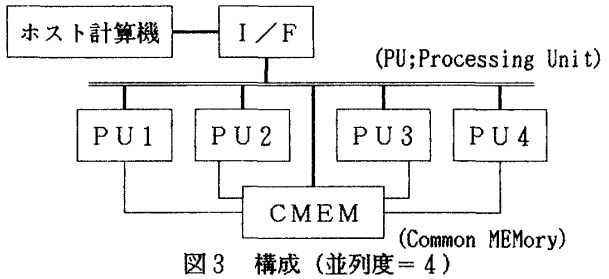

図３　構成（並列度＝４）

High-Speed Processing Method for the Morpheme Extraction Algorithm.

Osamu NAKAMURA, Akimichi TANAKA, Hideo KIKUCHI
NTT Communications and Information Processing Laboratories.

(2) 評価パラメータ

以下に示すように，アルゴリズムの高速性に大きな影響を与えると予想された入力パラメータを設定し，出力パラメータとの関係を評価

入力パラメータ	出力パラメータ
・入力文字列長	・処理時間
・並列度	・インデックスデータ量
・辞書規模	・抽出形態素個数
・階層分割レベル	

4．評価結果

形態素抽出処理時間Tは，式(1)で表わされる．

$$T = Ti + Td + Tio \quad \cdots(1)$$

Ti ；インデックス探索時間
Td ；辞書データの読出し時間
Tio；データの入出力時間

本処理方式によれば，式(1)中の時間要素の内，Tiについて木構造インデックスによる高速化効果が，また，TiとTdの両者について並列化の効果が期待できる．

以下，評価実験から得た効果検証結果を以下に要約する．

(1) 木構造インデックスによる高速化効果

並列度=1とした本処理方式の実行時間THとソフトウェアによる実行時間TSの比較から，机上評価で予測した1桁の高速化効果を実現可能であることを確認した(図4)．

(2) 並列化効果

式(2)で示される並列化効果が得られる(図4)．

並列化効果＝並列度n×（1－α） ・・・(2)

αは，抽出形態素個数のPU間の偏り(後述)に起因するオーバヘッドで，一般的にはnに依存するパラメータであるが，nが4までの範囲では，0.2～0.4であることが分かった．さらにnが5～20の範囲でも，CMEMでのアクセス競合の発生状況より，上記結果に大きな変化はないと予想される．

(3) 辞書規模の増加に対する実行時間の削減効果

大規模辞書を扱う場合の形態素抽出処理時間Tは，一般にその規模Dに比例して増加するが，本処理方式では木構造インデックスの特徴から

$$Ti \propto D^{1/4},\ Td \propto D,\ Tio \propto D \quad \cdots(3)$$

という結果が得られた．各時間要素の実測値から，THは，10万語以下の辞書では他方式の2/3に，それ以上の大規模辞書では1/3～1/2に削減されるものと見込まれる．

(4) 階層化によるインデックスデータの削減効果

階層分割レベル1～8における実行時間には有意差が認められなかったこと，インデックスデータ量が表1に示す傾向を有することから，階層分割レベルを1～2とすることで，上記(2)の並列化効果を低下させることなく，インデックスデータ量の増加を10%以内とすることが可能である．

表1　インデックスデータ量（並列度＝4）

LD	1	2	3	4	5	6	7	8
RD	1.0	1.1	1.5	2.3	3.1	3.6	3.9	4.0

LD;階層分割ﾚﾍﾞﾙ，RD;ﾃﾞｰﾀ量の比(LD=1を基準)

(補足)LD=3～5での急増は，2漢字（読みで4音節）の単語が最も多いという日本語の特性に起因するためと考えられる．

(5) 入力文字列長が処理時間に与える影響

PU間の形態素候補出力数の偏りは，入力かな文字列が長くなるとともに減少することから(*)，上記(2)のオーバヘッドαの削減には，処理対象文字列を長く設定することが並列化効果の改善に有効である．

(*) 出力データ量の最大／最小比で10字の場合の約10倍から70字の場合の約2倍へ減少

(6) データ入出力時間の比率

全実行時間に占める，データ入出力に要する時間の比率は高々15%であったことから，データ転送ネックの問題は生じないと考えられる．

5．あとがき

形態素候補抽出の高速実行を狙った処理方式について，辞書およびかな入力文字列に実際のデータを用いた評価実験を行い，高速化効果等の検証結果を示した．その結果，従来のインデックステーブルを用いるソフトウェアによる実行に比べ，約1桁，さらに並列度の0.6～0.8倍の並列化効果が得られることを示した．

また，抽出形態素個数を，複数PU間でバランスさせることによって，さらに並列化の効果を向上可能であることを示した．

[1]雪下他,33回情処全大p.1803,1986, [2]雪下他,34回情処全大p.1299,1987, [3]雪下他,情処計算機ｱｰｷﾃｸﾁｬ研究会66-4,1987, [4]中村他,COMPDEC'88,pp.488-495

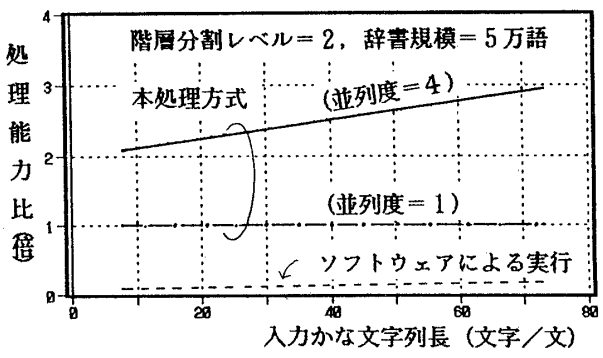

図4　処理能力の向上度