自然言語処理 121－9
情報学基礎 47－1
（1997. 9. 11）

形態素解析を用いた中間部分一致検索の高速化手法

奥 雅博、野田良輔、林 智定
NTT情報通信研究所
{oku, noda, tomo}@isl.ntt.co.jp

概要

本論文ではあらかじめ形態素解析を用いて階段状にインデックスレコードを派生させておくことによって、検索時の中間部分一致を高速化する手法について述べる。本手法は、PB入力型電話番号検索実験システムにおけるデータベース検索を高速化する目的で検討を進めてきたものである。本論文は、全文検索の分野でのPAT木に着目し、各レコードごとにPAT木における半無限部分文字列(sistring)に似た階段状のインデックスレコードを派生させ、それをDBMSで管理することによって、各レコードの中間部分一致を高速化するものである。評価実験の結果、本手法は、速度面、性能面、精度面で番号検索のタスクに要求される条件をかなりのレベルで満足することが検証できた。

Generating Supplementary Index Records for High-Speed Partial Matching Using Morphological Analysis

Masahiro Oku, Ryousuke Noda and Tomosada Hayashi
NTT Information and Communication Systems Laboratories

Abstract

This paper proposes a method for generating supplementary index records for high-speed partial matching using morphological analysis. We have been developing a fully automated directory assistance system using a telephone keypad. A telephone directory retrieval task requires (1) high-speed partial matching, (2) reducing resources such as CPU power and memories, (3) invoking many processes concurrently. We try to realize high-speed partial matching by modifying sistrings (semi-infinite strings) in PAT trees that is used in the full text search domain. The experimental results show that the proposed method satisfies the requirements mentioned above pretty well.

1．はじめに

我々はプッシュボタン(PB)信号送出可能な電話機を入力端末として利用するPB入力型電話番号検索実験システムの開発を進めている（東田 1997）（林他 1997）。このシステムは、家庭やオフィスに普及しているPB信号送出可能な電話機を用いて住所や名前の入力を可能とする日本語入力方式（佐藤他 1997）をもとに、入力情報を用いてデータベースを検索する技術、HMIに関連する対話誘導技術（奥他 1997）、および音声応答技術などから構成されている。電話番号検索では、利用者が入力したキーワード（検索キー）と一致するあるいは検索キーを一部に含む情報を効率よくしかも高速に検索することが要求される。一例として企業名の一部が入力された場合を考えてみよう（図1）。

図1に示すように、企業名「日本電信電話株式会社」を検索する際に、利用者が名義を完全に知っていればデータベースのインデックスとの完全一致を行えばよい。しかし、名義を完全に知っている利用者ばかりとは限らない。このような場合には、検索文字列として「日本電信電話」、「電信電話」、「電信」＆「電話」などが入力されることになる。これらの文字列で企業名「日本電信電話株式会社」を検索するには、図1に示すように中間部分一致検索を行わ

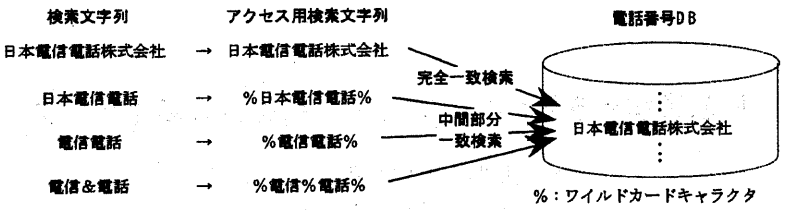

図１：　電話番号検索の例

なければならない。すなわち、一般には完全一致検索を行えることはまれであって、中間部分一致検索が基本操作となる。電話番号検索タスクでは、この操作を非常に大きなデータベースに対して高速に行うことが要求される。

上記の内容を含めて、電話番号検索タスクには以下のような要求条件がある；

（1）中間部分一致：　利用者からの不完全な入力に対しても何らかの解を求めるため（再現率を高めるため）に、検索キーとこれを含むインデックスとを照合させる部分一致検索を行う必要がある。電話番号検索では、インデックスは住所や企業名などのように固定しているが、レコード数は数百万から数千万件のオーダとなる。このような大量のレコードのインデックスと検索キーとの照合を高速に行う必要がある。

（2）高速性：　レスポンスタイムが重要である。このためには上記の中間部分一致検索を高速で行うことが必要である。DBMSの機能では、中間に位置する文字列に対してはインデックスが有効に働かないため、インデックスレコードの中間に位置する文字列と検索キーとの中間部分一致検索を高速で行うことはできない。

（3）多重化に耐えられること：　電話番号検索では１つのマシンで多数のプロセスを処理しなければならない。従って、重要なのは１プロセスでリソース（特にCPUパワーとメモリ）を使いすぎないこととディスクI/O量が大きくなりすぎないことである。

（4）近接検索：　名義が部分的にしかわからない場合には、複数の検索キーに一致（あるいは検索キーを包含）するインデックスレコードを取得する必要がある。DBMSでは、複数の検索キーの間にワイルドカードキャラクタを挟むことによって実現することができる。

本論文では、これらの要求を考慮して市販のDBMSを用いて、かつ、中間部分一致を高速に行う手法について提案する。本手法は、DBMSの持つインデックスを利用した検索の高速性を損なうことなく、中間部分一致を行えるようなインデックスレコードを派生させる。この派生を行う際に日本語形態素解析を利用する。

２．用語の定義

○検索キー

データベース検索を行う際の照合文字列。

○インデックス、インデックスレコード

ここでは、インデックスレコードの集まりをインデックスと呼ぶこととする。インデックスレコードは名義部とポインタ部とから構成される。名義部はデータベース検索を行う際に検索キーと照合される被照合文字列であり、ポインタ部はデータベース本体情報を指し示す。以下、インデックスレコードの名義部を単にインデックスレコードと呼ぶこととする。

○部分一致

検索キーの一部がインデックスレコードに含まれている場合、そのインデックスレコードに“部分一致”したという。例えば、検索キー＝「電話」は、インデックスレコード「日本電話」、「電話会社」や「電信電話会社」に部分一致する。また、検索キー＝「京都」は、インデックスレコード「京都府」の他に「東京都」のように単語境界をまたぐものにも部分一致する。このような部分一致は検索の適合率の低下を招く１つの要因となる。

○前方部分一致

部分一致のうち、検索キーがインデックスレコードの先頭から一致しているものを“前方部分一致”と呼ぶ。検索キー＝「電話」は、イン

デックスレコード「電話会社」に対して前方部分一致する。

○中間部分一致

部分一致のうち、検索キーがインデックスレコードの先頭ではない部分から一致しているものを“中間部分一致”と呼ぶ。検索キー＝「電話」は、インデックスレコード「日本電話」や「電信電話会社」に対して中間部分一致する。

○適合率

情報検索の指標の１つで、ある検索条件によって得られた結果の中に含まれる正しい結果の割合を適合率という。

○再現率

情報検索の指標の１つで、ある検索条件によって得るべき結果（正しい結果）のうち、実際に得られた結果の割合を再現率という。

３．従来の中間部分一致検索

従来のDBMSによる中間部分一致検索は、インデックスを持っている場合でも、インデックスレコードに対する前方からの照合ではないために、インデックスを用いた場合の検索の高速性を利用することはできず、インデックスを用いない文字列照合アルゴリズムである BF(Brute-Force)法、KMP(Knuth-Morris-Pratt)法、BM(Boyer-Moore)法、AC(Aho-Corashic)法などを用いて、すべてのインデックスレコードと検索キーとを照合する方法が採られている（Frakes 1992）。これらの方法では、レコード数が数百万から数千万件のデータベース検索においては、十分な速度で照合を完了し、中間部分一致検索を終了することはできなかった。

また、日本語文書の全文検索の分野では、全文からインデックスを作成する手法において、検索の高速化のために、文字成分表を用いた手法、n文字インデックスを用いた手法（赤峯他 1996）などが提案されている。これらの手法では、入力文字列を単語分割してそれらの単語を検索キーとして使用するため、入力文字列の解析が必要となる。また、これらの手法は高々数語の長さしかない項目をインデックスとするデータベースを検索する際の高速化手法として用いることは難しいと考えられる。

PAT木を用いた全文検索では、原文の文字列のすべてを始点とする半無限部分文字列(sistring)に対して、インデックスを張る方法を採用することによって、入力文字列と検索対象の全文字列との照合を可能としている（菊田 1996）（Frakes 1992）。

本論文で提案する手法は、各レコードごとにPAT木における半無限部分文字列に似た階段状のインデックスレコードを派生させ、パトリシアツリーを用いる代わりに、それをDBMSで管理することによって、各レコードの中間部分一致を高速化するものと捉えることができる。

４．形態素解析を用いた中間部分一致検索の高速化

４．１　基本的な考え方

DBMSにおいて中間部分一致検索が遅い要因の１つは、インデックスが有効に働かないことにある。部分一致の中でも前方からの部分一致検索（前方部分一致検索）に関しては、インデックスが有効に働くので検索は速い。すなわち、中間部分一致検索を前方部分一致検索に置き換えることができれば、検索速度を上げることができる。本論文では、元のインデックスレコードから中間部分一致検索を前方部分一致検索に置き換えるためのインデックスレコードを派生させることによって中間部分一致検索の高速化を実現する手法について提案する。

４．２　インデックスレコードの派生方法

中間部分一致検索を前方部分一致検索に置き換えるために、検索キーとなりうる文字列すべてを、新たなインデックスレコードとして元となるインデックスレコードから派生させる。このとき、以下の点に留意する（図１参照）；

（１）検索キーが１単語から構成されているとは限らない。

例）検索キー＝「電信電話」

→　２単語から構成されている。

（２）検索キーが１単語とは限らない。

例）検索キー＝「電信」＆「電話」

→　両単語を含むものを検索。

本論文で提案する手法は、各レコードごとにPAT木における半無限部分文字列に似た階段状のインデックスレコードを派生させることによっ

てこれらの点を解決するものである。

図２に本手法におけるインデックスレコードの派生方法について示す。まず、元となるインデックスレコードを形態素解析し、単語ごとに分割する。次に、インデックスレコードの先頭から単語を１つずつ削った単語列を新たなインデックスレコードとして派生させる。すなわち、図２に示すように元のインデックスレコードABCD（A、B、C、Dはそれぞれ単語）から、BCD、CD、Dという３つのインデックスレコードを派生させる。このとき、ストップワードと同じ考え方に基づいて、データベース中に高頻度で出現するものは派生させない。これは、高頻度のものは派生させても検索にとって意味をなさない（インデックスとして意味をなさない）からである。図２に示す派生させたインデックスレコードを用いれば、検索キーが１単語から構成されているか否かにかかわらず、検索キーを前方に含むインデックスレコードを検索（例：BC → BC%で検索（%はワイルドカードキャラクタ）、B → B%で検索）することにより、インデックスを有効に働かせながら所望の検索結果を得ることができる。また、検索キーが１単語でない場合にも、それらの検索キーのうちの一方を前方に含み、その後ろに他方の検索キーを含むインデックスレコードを検索（例：BとC → B%C%またはC%B%で検索）することにより、インデックスを有効に働かせながら所望の検索結果を得ることができる。

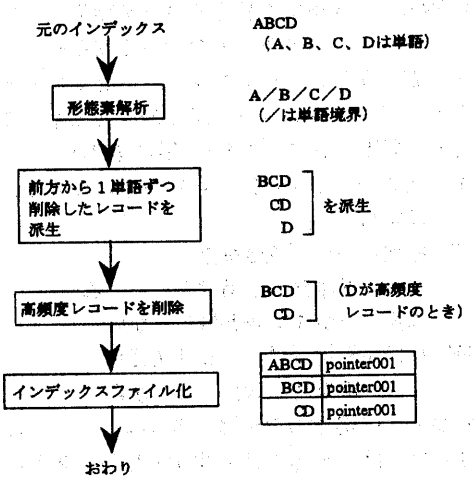

図２：　インデックスレコードの派生方法

４．３　インデックスレコード派生とこれを用いた中間部分一致検索の具体例

図３にインデックスレコード派生とこれを用いた中間部分一致検索の具体例を示す。

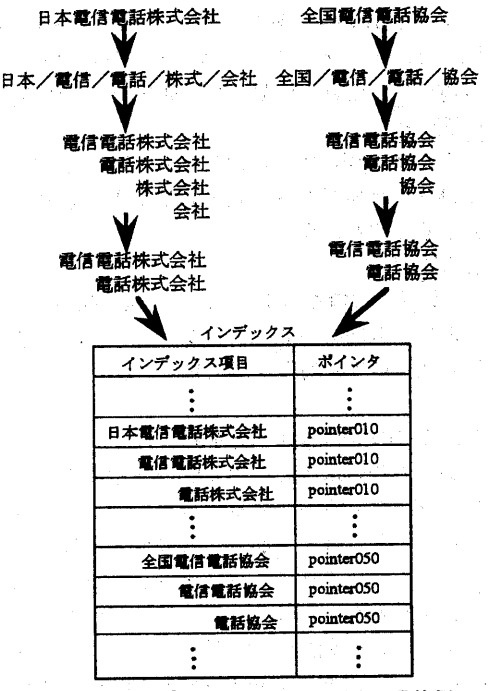

図３：　インデックスレコード派生の具体例

元のインデックスレコード「日本電信電話株式会社」を形態素解析すると、「日本／電信／電話／株式／会社」（／は単語境界）であるので、これから「電信電話株式会社」、「電話株式会社」、「株式会社」、「会社」の４つが派生対象となる。ここで、「株式会社」、「会社」の２つは高頻度に現れるため、インデックスレコードとしては派生させないものとする。従って、「電信電話株式会社」、「電話株式会社」の２つをインデックスレコードとして派生させる。「全国電信電話協会」の場合も同様にして「電信電話協会」、「電話協会」の２つをインデックスレコードとして派生させる（「協会」は高頻度とする）。

次に、上記のようにして作成したインデックスを利用した中間部分一致検索の例を示す。

（例１）「日本電信電話株式会社」での検索

「日本電信電話株式会社%」として検

索することによって、元のインデックスレコード「日本電信電話株式会社」と一致する。

（例２）「日本電信電話」での検索
「日本電信電話%」として検索することによって、元のインデックスレコード「日本電信電話株式会社」と前方部分一致する。

（例３）「電信電話」での検索
「電信電話%」として検索することによって、２つの派生インデックスレコード「電信電話株式会社」と「電信電話協会」に前方部分一致する。この結果、「日本電信電話株式会社」と「全国電信電話協会」とを検索結果として得ることができる。

（例４）「電信」＆「電話」での検索
「電信%電話%」として検索することによって、（例３）と同じく、２つの派生インデックスレコード「電信電話株式会社」と「電信電話協会」とが前方部分一致し、「日本電信電話株式会社」と「全国電信電話協会」とを検索結果として得ることができる。

（例１）、（例２）はともにインデックスレコードを派生せずとも前方部分一致で検索することが可能な例である。（例３）、（例４）は、インデックスレコードを派生させない場合には中間部分一致でしか検索できない例であり、本手法でインデックスレコードを派生させることにより、前方部分一致検索することが可能となる例である。

以上のように、４．２節で述べた手法によって、インデックスレコードを派生させることによって、中間部分一致検索を前方部分一致検索に置き換えることができるので、中間部分一致検索を高速に行うことができる。また、電話番号検索のタスクに特徴的な、（１）検索キーが１単語から構成されているとは限らない、（２）検索キーが１単語とは限らない、についても対応することができる。

５．評価実験

本論文で提案した手法の有効性を確認するために、検索速度面、性能面から見た評価実験を行った。

５．１　方法

５．１．１　実験１

市販のDBMSであるOracle 7.3.2.2を使用して以下の３つの方法で検索速度を測定した；

（１）%key%による中間部分一致検索（インデックスを使用しない場合）。

（２）提案手法によりインデックスレコードを派生させた後のデータベースに対するkey%による前方部分一致検索。

（３）DBMSが用意している全文検索（Oracle 7.3.2.2のコンテキストオプションによる検索）。

このとき、CPU使用率、ディスクI/O量、メモリ使用率についても計測を行った。検索対象としては、PB入力型電話番号検索実験システムで使用している企業名DBのうち、札幌地区の部分を用いた。この企業名DBは、電話帳に掲載されている企業名と電話番号とから作成した。

検索キーとしては、ランダムに発生した４バイト文字列（４文字のかなに相当）のうち、（３）のコンテキストオプションによるヒット件数が１０～１００件程度となるもの１００件を用いた。これは、キャッシュにデータが乗ってしまって、異様にレスポンスが速くなることを避けるためと、実際にヒットするデータで、ある程度の負荷が均等にかかるようにするためである。レスポンスタイムとしては、Oracle 7.3.2.2でのSQL解析からすべてのヒットレコードを取得し終えるまでの経過時間を測定した。また、システム全体のCPU使用率、ディスクI/O量、メモリ使用率については、５秒間隔で測定した。計測時間を確保するため、（２）提案手法による検索と（３）コンテキストオプションによる検索の場合には１００件の検索を５回繰り返し、都合、５００件の検索時間を測定した。

５．１．２　実験２

より番号検索タスクに近い条件での検索速度を測定するために、提案手法とコンテキストオプションによる検索を、検索キーの長さの異なる５０件に対して実施した。検索対象としては、

企業名DBの北海道地区全域を用いた。検索実験は、キャッシュ効果を抑えるために、データベースを再立ち上げ直後に５０件の検索を行い、検索時間を測定した。また、コンテキストオプションによる検索では、ヒット件数が０件のときには単語分割による再検索を実施し、単語分割後の検索時間を測定した。

５．２　評価実験結果

５．２．１　実験１の結果

検索時間、CPU使用率、ディスクI/O量に関する実験結果をそれぞれ図４、図５、図６に示す。なお、メモリの使用率は３つの手法でほとんど差がなかった。

図４より以下のことがわかる；

（1）ヒット件数が少ない場合には、提案手法は非常に高速に検索結果を得ることができる。これはインデックスが有効に働いていることを示している。

（2）提案手法はヒット件数が増えると検索時間が増える。これは、ヒット件数が増えることによって取得する件数が増加することに起因していると考えられる。また、ヒット件数が多いものに対してはインデックスが有効に働かなくなるので、結果としてデータベースの多くの部分との照合が必要であるためと考えられる。

（3）コンテキストオプションによる検索においてもヒット件数の増加とともに検索時間も増加しているが、その増加の割合は提案手法による検索に比べて小さい。

（4）これに対して、中間部分一致検索はヒット件数によらず一定の時間で検索が可能であるが、検索時間は他の２方法に比べてかなり遅い。これは、単純な中間部分一致検索ではインデックスを使えず、データベースすべてとの照合を行わなければならないためであると考えられる。

（5）検索時間に関して、ヒット件数が約５０件以下では提案手法とコンテキストオプションによる検索とで同程度である。

また、図５、図６より以下のことがわかる；

（6）CPU使用率は提案手法の方が中間部分一致検索に比べて約２／５であり、コンテキストオプションによる検索に比べても約２／３である。中間部分一致検索のCPU使用率が高いのは、文字列の比較をすべてのレコードに対して行わなければならないためであると考えられる。

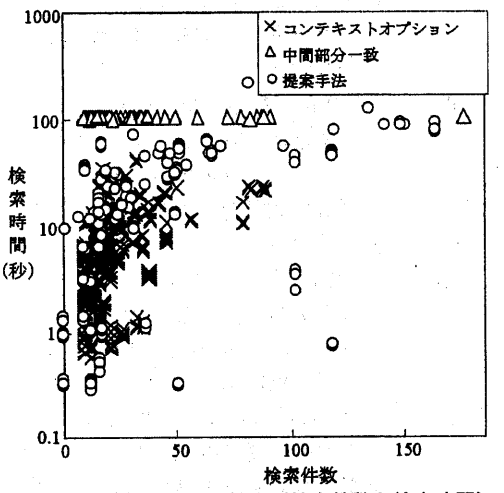

図４：　評価実験１の結果（検索件数と検索時間）

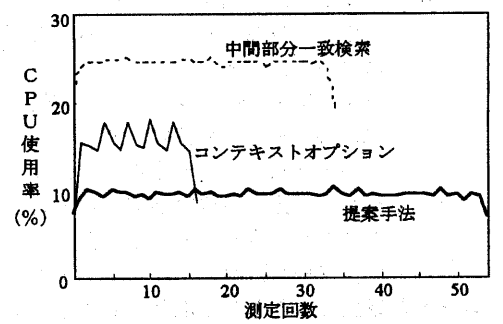

図５：　評価実験１の結果（CPU使用率）

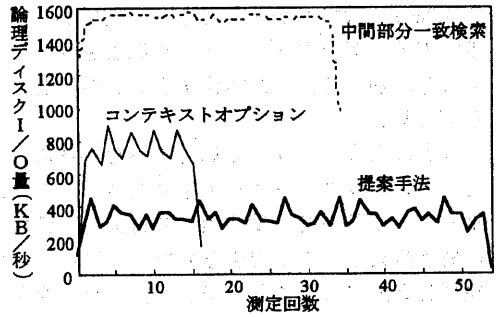

図６：　評価実験１の結果（ディスクI/O量）

（7）ディスクI/O量も、提案手法の方が中間部分一致検索に比べて約１／５であり、コンテキストオプションによる検索に比べても約１／２である。中間部分一致検索のディスクI/O量が大きい理由も、すべてのレコードを検索して文字列比較処理を行わなければならないためであると考えられる。

（8）提案手法のCPU使用率、ディスクI/O量は、他の２つの検索方法に比較してかなり小さい。これらのことは、多重度を上げる上で提案手法が有効であることを示している。

５．２．２　実験２の結果

実験２の結果を図７（検索件数と検索時間との関係）、図８（検索ごとの検索件数）に示す。

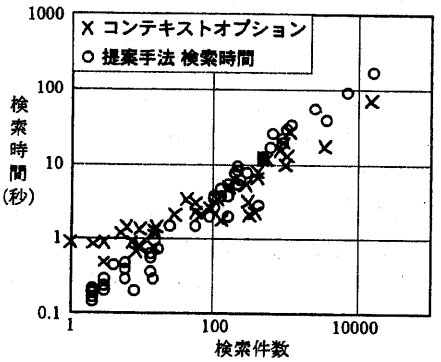

図７：　評価実験２の結果

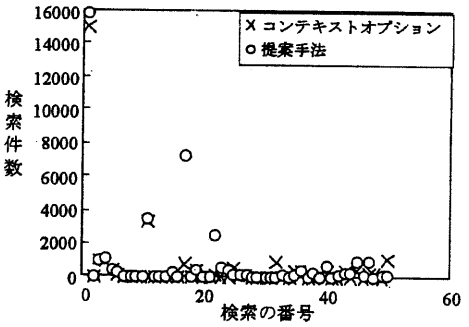

図８：　評価実験２の結果（検索件数）

図７より、両者の検索速度は同程度であると言える。特にヒット件数が少ない（約１００件以下）場合には提案手法の方が若干高速である。

図８より、コンテキストオプションによる検索の方が提案手法による検索よりもヒット件数が少ないものが多い。また、単語分割によってヒットしたものもあった（５０件中１９件）。これらのことはコンテキストオプションによる検索の方が提案手法による検索に比べて再現率が低いことを示している。また、番号検索というタスクからはヒットしないものがあるというのは非常に大きな問題である。これに対して、提案手法による検索では単語の先頭から始まるものに対する検索もれはなく、再現率の点からは問題ないと考えられる。

５．３　考察

速度面から見ると、提案手法はコンテキストオプションと同等の検索速度を得ることができると考えられる。特にヒット件数が小さい場合には提案手法による検索の方が若干高速であると推定される。

性能面から見ると、提案手法による検索は中間部分一致およびコンテキストオプションによる検索に比べてディスクI/O量も少なく、CPU使用率も低い。これらのことは全体の処理のスループットが高いことを示しており、番号検索タスクの１つの要求条件である多重度を上げるという点を満たしている。また、インデックスを派生させることによって、インデックスやテーブル等のディスク容量が余分に必要となるが、近年のディスクの安さ、手に入りやすさを考えると大きな問題とはならないと考えられる。

精度面から見ると、コンテキストオプションによる検索ではヒットしない場合や単語分割を行わないとヒットしない場合が存在するため、再現率が下がるという問題がある。すべての全文検索がこのような結果になるとは限らないが、日本語文書に対する全文検索が単語を単位に構築されていることから、再現率を上げるためには検索キーを単語分割する必要があるというのは共通であると考えられる。すなわち、全文検索では入力された検索キーを単語分割した後にデータベースを検索する必要がある。しかし、入力された検索キーをリアルタイムに形態素解析して単語単位に分割することは多重度が高いことが要求される番号検索タスクには合わない。

次に、適合率の向上策について検討する。単純な中間部分一致検索では、単語の途中から始

まる文字列と検索キーとが一致した場合、それが検索のノイズとなって適合率を下げる要因となってしまう。検索の後、単語境界を見て、余分な候補を削除する方法を採る必要がある（永井　1997）。提案手法では単語境界をインデックス作成に利用しているので、単語境界位置の情報を候補のチェックに利用することによって、不要な候補（単語境界とは異なる場所で切れている候補）を削除することができ、結果として適合率を向上させることができる。

以上の点から、番号検索というタスクに対しては、高速性という面の他に性能面、精度面から見て、提案手法による中間部分一致検索の高速化は非常に有効な手段であると結論できる。

6．おわりに

本論文ではあらかじめ形態素解析を用いて階段状にインデックスレコードを派生させておくことによって、検索時の中間部分一致を高速化する手法について述べた。さらに評価実験を実施してその有効性を検証した。本手法はPB入力型電話番号検索実験システムのデータベース検索に使用されている（林他 1997）。本システムの日本語入力は、PB電話機を用い、１押下で１文字を入力する方式のため、入力に曖昧さが残る（佐藤他　1997）。このため、入力を形態素解析して単語単位に分割することは困難である。従って、データベース検索では１単語相当の文字列入力、複数単語からなる文字列入力にも対応しなければならない。本論文で提案した手法はこれらの要求を満たすことができる。評価実験の結果、速度面でも精度面でも多重度（本論文では触れなかったが、CPU使用率、メモリ使用量から見て、１クライアント当たり約５０回線分の処理が可能であるという結果を得ている）という面でもかなりのレベルで要求条件をクリアしていることがわかった。

[参考文献]

（赤峯他　1996）赤峯　享、福島俊一、高速全文検索のためのフレキシブル文字列インバージョン法、Proc. of Advanced Database Symposium '96, pp.35-42 (1996).

（Frakes 1992）Edited by William B. Frakes and Ricardo Baeza-Yates, Information Retrieval, Prentice Hall PTR (1992).

（林他　1997）林　智定、東田正信、佐藤　亨、ＰＢ電話機を利用した電話番号案内技術、97信学会総合大会、D-6-5 (1997).

（東田　1997）Masanobu Higashida, A Fully Automated Directory Assistance Service that Accommodates Degenerated Keyword Input Via Telephone, Proc. of PTC'97, pp.167-174 (1997).

（菊田　1996）菊田昌弘、用語解説：パトリシアツリー(Patricia Tree)、人工知能学会誌、Vol.11, No.2, pp.337-339 (1996).

（永井他　1997）永井良史、林　智定、野田良輔、文字区切り・単語区切りを用いた検索解の絞り込み効果の検討－ＰＢ電話機を利用したデータベース検索への応用－、97信学会総合大会、D-6-8 (1997).

（奥他　1997）奥、林、永井、東田、“PB電話機を利用した電話番号案内方式に適した対話誘導戦略”、97信学会総合大会、D-6-7 (1997).

（佐藤他　1997）佐藤、東田、林、奥、村上、“PB電話機を利用した日本語入力方式”、97信学会総合大会、D-6-6 (1997).