情 報 学 基 礎 25－4
（1992. 5. 12）

ソフトウェアによる テキストサーチマシンの実現

畠山 敦　浅川悟志　加藤寛次
（株）日立製作所

大容量の文書データベースを対象として、既に開発した高速全文検索システム“テキストサーチマシン（ＴＳＭ－Ｉ）”をベースに、今回比較的小規模の文書ＤＢを対象にワークステーション上のソフトウェアで動作するフルテキストサーチシステム（ＴＳＭ／ＥＭ）を開発した。本システムでは、ソフトウェア型の低速性を改善するために文字の連続情報を利用した連接文字成分表サーチ方式によるプリサーチと、高速文字列照合ハードウェアをエミュレートするサーチエンジン・エミュレータを開発することによって、単純条件検索時に１．５～１２ＭＢ／ｓの等価検索速度を得ることができた。

Development of A Software Full-Text Search Machine

Atsushi Hatakeyama　Satoshi Asakawa　Kanji Kato
Central Research Laboratory, Hitachi,Ltd.
Kokubunji, Tokyo 185, JAPAN

The development of a full text search system (TSM/EM) is discussed. This system is based on TSM-I which was previously developed as a prototype. TSM/EM does not require specialized hardware. It is designed to run on a workstation. The pre-search method uses a character bitmap table which registers many pairs of adjacent characters in database text. It achieves very fast retrieval of documents. A search engine emulator, which emulates fast string matching hardware, has been developed. These methods have achieved 1.5 - 12 MB/s system search speed for simple condition queries.

1．はじめに

近年、ワードプロセッサやパーソナルコンピュータ、ワークステーションなどの普及拡大に伴い、作成される文書情報も急速に増加してきており、近い将来膨大な量に達するものと予想されている。このため、大量の文書情報を一般のユーザが簡単に蓄積検索できる文書情報検索システムに対する要求が高まりつつある。また、既存システムにおいても、文書データベースの大規模化に伴う絞り込み率の低下や、技術文献データベースでの技術用語の目まぐるしい変遷に起因する検索精度の低下などが大きな問題となってきている。

こうした要求や問題に応えるため、インデクス情報を用いない自由な言葉による検索を目的としたフルテキストサーチ（自由語全文検索）技術の研究を行ってきている。本研究の中で、その実用性を確かめるため、既にテキストサーチマシン（ＴＳＭ－Ｉ）を試作し、発表してきた。[1]

今回は、そのプロトタイプの技術を活かしソフトウェアのみで動作する、ワークステーション３０５０ベースのフルテキストサーチシステムを開発することができたので、報告する。

2．システムの概要

2.1　特　長

今回開発したシステムは、以下の特長を持つ。

（１）階層プリサーチ方式（連接文字成分表方式）

階層プリサーチはフルテキストサーチを高速に行うために、対象テキストを順次絞り込んでいく方式である。特に、今回この絞り込み率を向上させるために、テキスト中の連接する２文字を単位に文字成分表を作成する方式を開発した。これにより、絞り込み率のばらつきを抑え探索量を減らすことができ、検索時間を短縮することができた。

（２）サーチエンジンエミュレート

ＴＳＭ－Ｉにおいてハードウェアで実現していた文字列照合処理を、ソフトウェアでエミュレートし、コストの低減を図っている。この照合処理には、複数個の検索タームを一度のテキストスキャンで探索するアルゴリズムを用いている。

（３）同義語異表記展開

フルテキストサーチの課題である検索もれを解消するため、ユーザの指定した検索タームと同義なタームを同義語辞書を参照して展開しそれらも検索タームとして探索する方式としている。また、外来語などのカタカナ表記のゆれは、部分文字列の置き換え規則を作成し、これを用いて異表記を生成し探索する方式としている。

（４）複合条件処理

単純に指定された検索タームが存在する文書を抽出する機能の他に、検索ターム間の論理条件や、テキストデータ内での検索タームの出現位置関係を条件として指定して検索する機能を持っている。この機能により、きめの細かな検索が行えるため、的確に目的の文書を探し出すことができる。

（５）クライアントサーバシステム

このシステムは、ＴＣＰ／ＩＰプロトコルでの検索サーバとして機能する。したがって、ネットワーク上の他のワークステーションからも文書検索を行うことができる。

2.2　構成及び仕様

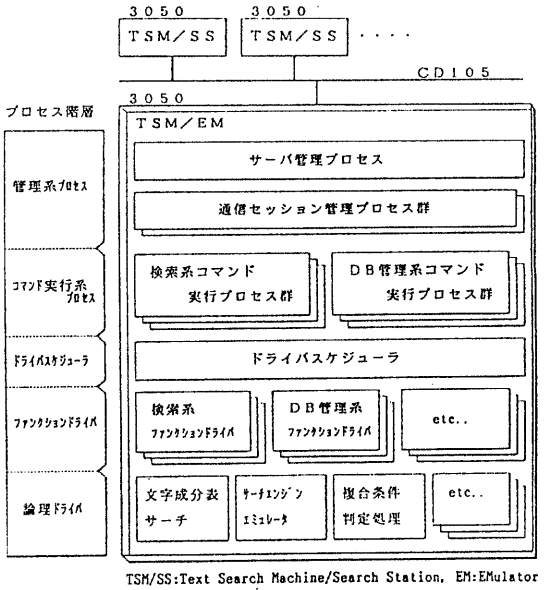
図１に今回開発したシステム（ＴＳＭ／ＥＭ）のソフトウェア構成を示す。このソフトウェアは、以下のような階層で構成されている。

（１）管理系プロセス

サーバとしての通信と各種の実行プロセスを管理するプロセスを管理系プロセスと呼ぶ。全体を管理するサーバ管理プロセスと、ほかのＷＳからの検索要求を受け付け、コマンド実行プロセスを起動する通信セッション管理プロセスからなる。通信セッション管理プロセスは、他の検索ステーションから接続要求があるたびに起動され、それぞれが要求検索ステーションと通信し、コマンドを受け付ける。

（２）コマンド実行系プロセス

検索条件式をサーバに伝えたり、検索モードを設



TSM/SS:Text Search Machine/Search Station, EM:EMulator

図１　ＴＳＭ／ＥＭソフトウェア構成概要

定する検索系コマンドを実行するプロセスと、ＤＢ作成とメンテナンスを行うＤＢ管理系のユティリティコマンドを実行するプロセスがコマンド実行系プロセスである。これらのプロセスは、実行要求が来る度に起動され、処理を実行する。検索系コマンドは、通信プログラムを介して、検索ステーション（クライアント）から送られてくる。このコマンドを通信セッション管理プロセスが受け、対応するコマンド実行プロセスを起動する。ユティリティコマンドは、今回サーバ側でのコマンド入力で起動するかたちにしている。そのため、ユティリティ系コマンド実行プロセスは、サーバ側シェルで直接実行する方式としている。

（３）ドライバスケジューラ

コマンド実行系から送られる各種の要求を受け付け、ファンクションドライバを起動するのがドライバスケジューラである。サーチエンジンなどハードウェア・アクセラレータを搭載した際のリソース管理をするための排他制御の機能も備えている。排他制御は、待ちキューバッファを用いて行い、リソースの空きが無い場合には、キューに要求を溜めるようにしている。

（４）ファンクションドライバ

コマンド実行系で細分化された処理を実行するのがファンクションドライバである。具体的には、ＤＢ操作、検索処理、ファイル情報取得などの機能を持たせている。これらのファンクションをコマンド実行系プロセスが組み立てて、検索処理やＤＢ構築処理を実現する。

（５）論理ドライバ

基本的な処理あるいは将来ハードウェア・アクセラレータに置き換え得る処理を、論理ドライバとして最下層に置いた。例えば検索時には、ディスクからデータを読み出すＭＤＥ論理ドライバ、文字列照合動作を行うＳＥＥ論理ドライバ、複合条件判定を行うＣＣＵ論理ドライバの各論理ドライバが起動され、セマフォにより同期して動作し、検索処理が行われる。

以上のソフトウェア構成により、表１から表３に示す機能を実現した。機能は、検索系、ＤＢ構築系、およびそれ以外のユティリティ系に分けて示している。検索系機能は、サーチコマンドが文字列として検索ステーションから通信プログラムを介してサーバへ送られ、実行されることで実現される。すなわち、この機能は検索サービスを受ける一般ユーザのために提供される検索機能である。ＤＢ構築系およびユティリティ機能は、サーバ上で直接コマンドを入力して起動される。これらは、ＤＢ管理者が保守のために使用する機能である。

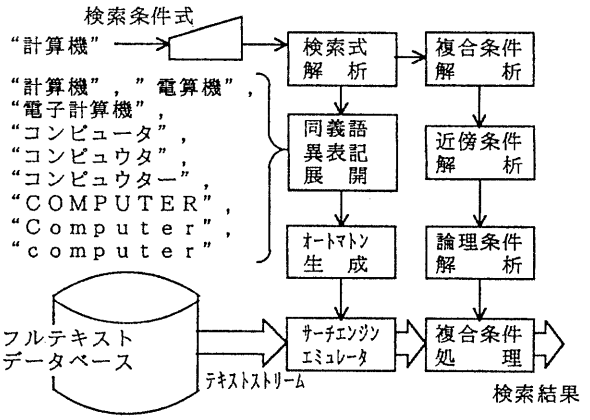


図２　検索処理の流れ

３．検索処理の概要

３．１　検索処理の流れ

今回のシステムにおける検索処理の流れを図２に示す。キーボードから入力された検索条件式は、まず検索タームと、検索ターム間の拘束条件を示す複合条件に分解される。検索タームについては、同義語異表記展開処理を行って、検索タームと同義なタームと外来語などのカタカナの異表記に関して異なるタームに展開する。次に展開後の検索タームを基にサーチエンジン・エミュレータを動作させるためのオートマトンを生成する。ここでは、同義語異表記展開処理で得られた検索タームのすべてを探索するようにオートマトンが生成される。

一方、複合条件はテキスト中における検索ターム間の位置条件としての近傍条件と、検索ターム間のＡＮＤやＯＲ条件を表す論理条件に分解して、複合条件判定処理へ送られる。そして、ここでは指定条件に適合する文書だけが検索結果として出力される。

３．２　階層プリサーチ方式

階層プリサーチとは、検索タームの存在する可能性がない文書を読み飛ばすことで、フルテキストサーチの検索速度を加速する検索方式である。スキャンする必要のない文書かどうかを判定するために、テキストデータを単語レベルで圧縮した凝縮テキストと、文字レベルで圧縮した文字成分表を用いている。特に今回は、文字成分表として２文字単位で文書中の使用文字を登録する連接文字成分表による階層プリサーチを実現した。凝縮テキストは、文字種の変化点で文字列を分割し、ひらがな文字を削除し、

表１　検索系機能一覧

No.	機　　能	備　　考
1	ＤＢ情報出力	ＤＢ名、ＤＢ構造の出力
2	ＤＢ選択	対象となるＤＢの選択
3	語義表記展開検索	同義語や異表記を含めた検索 （異表記：ﾙｰﾙ〔ﾒｰｶ提供〕、同義語：辞書〔ﾕｰｻﾞ作成〕）
4	ﾊｲｱﾗｰｷ/ﾕﾆﾊﾞｰｽ検索	前回の検索結果集合に対する絞り込み検索
5	検索対象ﾌｨｰﾙﾄﾞ指定検索	検索の対象となるﾌｨｰﾙﾄﾞの指定
6	単純検索	単一の文字列指定による検索
7	論理条件検索	複数タームによるＡＮＤ、ＯＲ条件による検索
7	距離指定検索	二つの検索ターム間の文字距離を指定した検索
8	同一ﾌｨｰﾙﾄﾞ内指定検索	同一ﾌｨｰﾙﾄﾞ内での複数検索ﾀｰﾑの共起を指定した検索
9	出現順序指定検索	複数検索ﾀｰﾑの出現順序関係を指定した検索
10	照合件数出力	検索結果件数の出力
11	照合ﾃｷｽﾄ出力	検索結果テキストの出力
12	文書ID指定ﾃｷｽﾄ出力	文書ＩＤによるテキストの直接指示出力
13	語義表記展開結果出力	同義語異表記展開結果の出力
14	ｺﾏﾝﾄﾞ取消し	実行中コマンドのキャンセル

表２　ＤＢ構築系機能一覧

No.	機　　能	備　　考
1	ＤＢ構造定義	最大容量定義、ﾌｨｰﾙﾄﾞ構造定義、格納ﾃﾞｨﾚｸﾄﾘ定義
2	ＤＢ一覧出力	登録ＤＢ名の一覧出力
3	データ登録	文字成文表、凝縮ﾃｷｽﾄ、ﾃｷｽﾄの作成、登録
4	データ削除	登録データの無効化
5	コンデンス	ＤＢ中の無効データ領域の削除
6	ＤＢ再構築	最大容量の変更
7	ＤＢ追加	ＤＢの追加
8	ＤＢ削除	ＤＢの削除
9	ＤＢ管理情報出力	容量、更新日時等の出力

表３　ユティリティ系機能一覧

No.	機　　能	備　　考
1	同義語辞書作成	同義語展開に用いる辞書の登録
2	サーチマシン初期化	異常終了時等に残された一時ファイルの削除
3	サーチマシン消去	全ＤＢ、管理テーブル等の消去
4	ﾌﾟﾘｻｰﾁﾌｧｲﾙ格納媒体定義	ディスク／メモリの選択

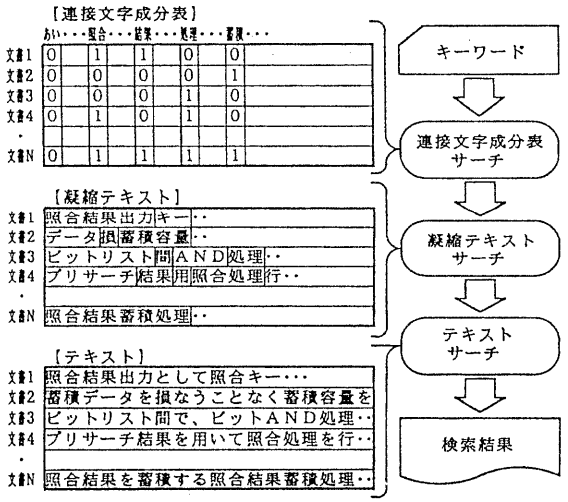


図３　階層プリサーチ方式

かつ重複する部分文字列を排除して圧縮したものである。

この階層プリサーチ方式を用いた検索処理の流れは、図３のようになる。まず、与えられた検索タームを２文字単位に分割し、文字成分表を用いて検索ターム中の全ての連接文字を含んでいる文書を候補として抽出する。次に、得られた候補文書についてその凝縮テキストをスキャンし、単語レベルでの照合を行う。詳細は３．４節の検索アルゴリズムで述べるが、必要に応じて、最後にテキストデータをスキャンして最終の検索結果を得る。

この方式により、高速でかつ検索もれのないフルテキストサーチを実現することができる。

３．３　サーチエンジン処理方式

ＴＳＭ／ＥＭにおける文字列照合処理には、オートマトン型の照合方式を採用している。この方式は、複数個の検索タームを一度のテキストスキャンで照合できるという特徴を持つ。これは、同義語異表記展開により増加した検索タームを高速に照合処理するのに必須なものである。

典型的なオートマトンの形を図４に示す。図は“計算機”という検索タームについて、同義語と異表記展開した後のタームを全て探索するオートマトンの例を表している。

３．４　検索アルゴリズム

検索処理のアルゴリズムを図５に示す。まず、第一にサーチコマンドとしてクライアントから送られてくる検索条件式の解析を行い、得られた検索ターム

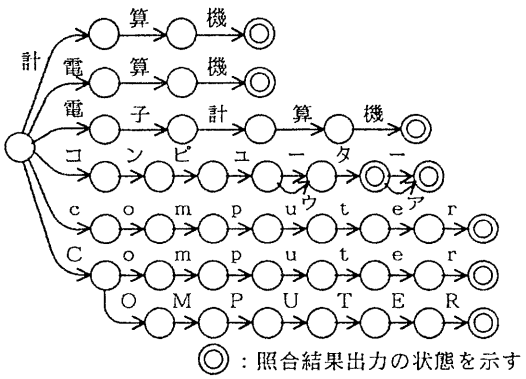


図４　文字列探索オートマトン

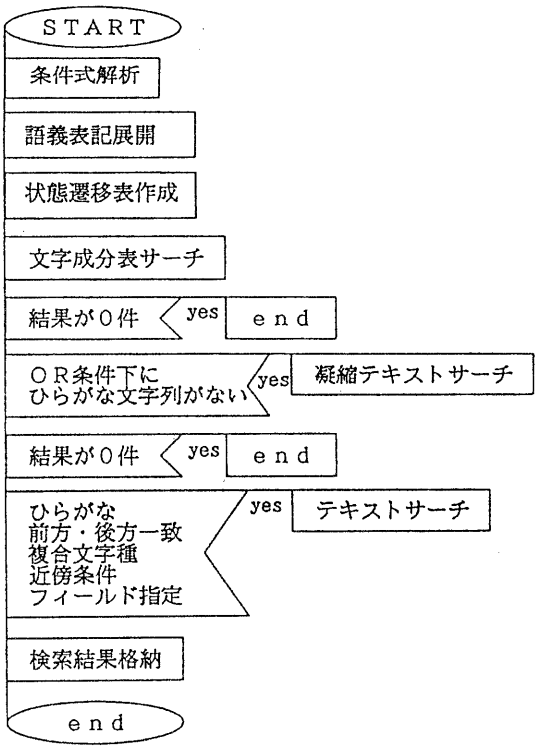


図５　検索実行アルゴリズム

ムを同義語異表記展開する。次に展開処理した検索タームを照合するオートマトンを作成する。

次に、指定された検索ターム中の文字をすべて含む候補文書を、文字成分表サーチで抽出する。ここでは、検索ターム中の連接文字単位に、連接文字成分表を参照し相互のビットＡＮＤ処理を行うことで、すべての文字を含む文書を抽出する。この文字成分

表サーチの結果が０件の場合には、次の階層プリサーチへ進む必要がないので、ここで検索処理を終了する。

この後で、凝縮テキスト探索を行う。凝縮テキストには、ひらがな文字列が含まれていないため、ひらがな文字列が検索ターム中に含まれる場合、凝縮テキストサーチをスキップする。例えば、単純条件で“あいまい”という文字列のみを探索する場合には、凝縮テキスト中でこの文字列が削除されているので、凝縮テキストサーチを行ってもヒットしないことになる。そのため、凝縮テキストをスキップしてテキストサーチを行い、正しい結果を得る必要がある。凝縮テキストサーチを行った結果が、文字成分表サーチのときと同様に０件になった場合は、ここで検索処理を終了する。

最後にテキストサーチを行う。ここでは、まず凝縮テキストサーチの結果で十分か否かの判定を行う。一例として、単純検索で“情報処理”という文字列を探す場合をあげる。“情報処理”という言葉を含むすべての文書は、凝縮テキスト中にその文字列を必ず含んでいる。したがって、テキストサーチを行う必要がなく、凝縮テキストの検索結果を最終結果として出力できる。一方テキストを探索する条件は、以下に示す通りである。

（１）検索タームにひらがなが指定されている場合
（２）近傍条件の指定がある場合
（３）文字種の異なる複合文字種タームがある場合

これらのいずれかの条件が成り立つ場合に限りテキストサーチを実行し、それ以外は凝縮テキストの検索結果を最終結果とすることで、処理を高速化している。

以上の検索処理の最終結果は、ヒットした文書の集合として、ファイルに格納し処理を終了する。この検索結果集合は、後に集合間論理演算処理とハイアラーキ・ユニバース検索で使用する。

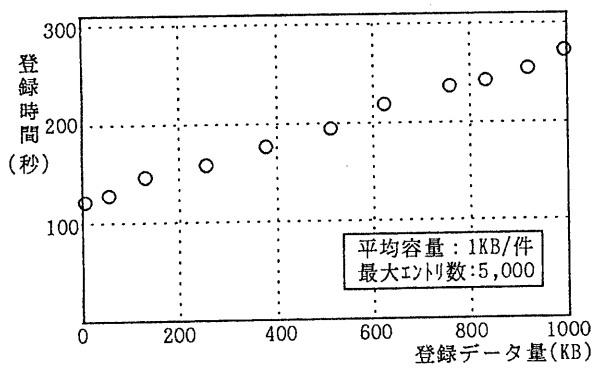


図６　１件当りの登録時間

また、検索要求を発行したクライアントＷＳへは検索結果件数が検索コマンドの応答として返送される。

４．性能評価

４．１　登録時間

図６にテキストデータの容量と登録時間の関係を示す。この図から、登録データの容量にしたがって、に処理時間が増加することがわかる。１０００件程度すなわち１ＭＢのデータ登録のとき、１件当りのデータ登録時間は、約２７０ｍｓである。

また、図でグラフが原点を通らずに最小容量でも１２０秒程度のオーバヘッドがみられるが、これは文字成分表の更新アルゴリズムによるものと考えられる。すなわち、ＤＢに登録できる最大文書数分の文字成分表を使って更新しているためである。例えば測定に用いた最大エントリ数が５，０００件のＤＢでは約３ＭＢのファイル操作が無条件に起きてしまっているため、１２０秒のオーバーヘッドが生じている。ただし、このオーバヘッドは一回の更新作業につき一度しか起きないので、まとめて登録するときには影響が小さくなり、１件当りの登録時間に換算すると非常に短くなる。

４．２　検索時間

階層プリサーチの傾向として、文字成分表サーチ結果の絞り込み率や、検索条件の特性に検索処理速度が大きく左右される。例えば、文字成分表サーチの結果、多くの文書がヒットするとそれだけ多くの凝縮テキストや、テキストデータをスキャンしなければならない。また、検索タームが単一の文字種であれば、凝縮テキストまでで検索処理が終了するのに対して、複合文字種の場合には凝縮テキストサーチで得られた候補文書に応じたテキストサーチを行う必要が生じる。

これらの要因から、検索処理時間を以下の４種類の条件下で測定した。

（１）単純条件

漢字あるいはカタカナによる単一文字種で検索タームを指定した場合には、凝縮テキストで検索処理が終了する。

（２）論理条件

漢字あるいはカタカナによる単一文字種で検索タームを指定し、かつ論理条件を指定した場合でも凝縮テキストで検索処理が終了する。ただし、検索ターム間の論理条件判定を行うため、単純条件よりも複合条件判定プログラムにかかる負荷が大きいので、検索速度が低下する。

（３）複合文字種

漢字とカタカナの組合せ、あるいは英字と数字の

組合せなど、異なる文字種での単純条件検索を行った場合、（１）と条件式は同じだが、内部的には凝縮テキストを検索タームの部分文字列で論理条件検索し、テキストを単純条件検索する。したがって、検索速度は、単純条件より遅くなることになる。

（４）近傍条件

検索ターム間のテキスト中の位置関係を指定した検索を行った場合、凝縮テキストを検索ターム間の論理条件で、テキストを近傍条件で検索することになる。最も処理の負荷のかかる検索条件である。

以上の条件での測定結果を図７に示す。検索時間は、検索クライアント側での待ち時間、すなわち検索レスポンス時間を示すものである。また、データベースには新聞記事２万５千件を用いた。テキストデータの容量は約２５ＭＢ、凝縮テキストの圧縮率は５０％である。図から、検索処理のオーバヘッドは約２秒であることがわかる。これは、通信と条件式の解析およびＴＳＭ／ＥＭ内部でのコマンド実行に伴うオーバヘッドであると考えられる。

単純条件と論理条件は凝縮テキストで処理が終了するため、複合文字種検索や近傍条件と比べ半分以下の時間となっている。複合文字種、近傍条件と単純条件、論理条件の間で顕著な差がみられないのは、複合条件判定プログラムにかかる負荷に較べて文字列探索処理のサーチエンジン・エミュレータにかかる負荷の方が、はるかに大きいことを表している。

また、凝縮テキスト止まりの検索に較べ、テキストまで処理が及ぶ場合には約３倍の時間を要している。これは、凝縮テキストであまり候補が絞られていないことを表している。すなわち、凝縮テキストの圧縮率を５０％とすれば、凝縮テキストサーチと同じ文書をテキストサーチした場合、凝縮テキストサーチの約２倍の処理時間がかかることになり、凝縮テキストサーチ時間を含めると全体で約３倍の処理時間になると言える。

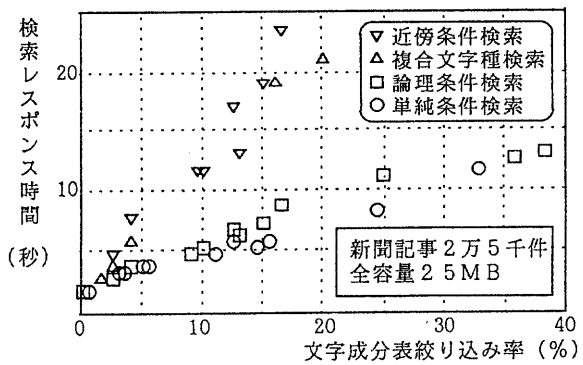


図７　システム検索性能

以上の結果から、文字成分表の絞り込み率が１０％程度のときは、単純条件あるいは論理条件検索時に、5′～６秒で検索処理が終了することが分かる。また、近傍条件あるいは複合文字種検索時には、１０～１５秒で検索が終了する。最も早く検索結果が得られるのは、文字成分表の結果が０件に近いときで、２秒で検索することができる。このことから、２５ＭＢのＤＢの検索処理速度は、１．５ＭＢ／ｓ～１２ＭＢ／ｓであるといえる。

５．おわりに

我々が開発したプロトタイプ、フルテキストサーチマシン（ＴＳＭ－Ｉ）の技術をベースに汎用ワークステーション３０５０上で、全てソフトウェアによるフルテキストサーチシステムを開発した。本システムは、

（１）テキストをスキャンする前に検索対象を高精度で絞り込む連接文字成分表方式

（２）複数個の検索タームの照合処理を一度のテキストスキャンで行うサーチエンジン・エミュレータ

（３）フルテキストサーチでの検索もれを解消する同義語異表記展開処理方式

（４）検索ターム間の論理条件や近傍条件を判定する複合条件判定処理方式

（５）ネットワーク上の他のワークステーションから検索要求が出せるクライアントサーバシステム

という特長を持つ。

また、検索処理速度は、典型例で１．５ＭＢ／ｓ～１２ＭＢ／ｓという性能が得られた。

参考文献

[1] 加藤,他,「大規模文書情報システム用テキストサーチマシンの研究」,情報学基礎14-6 (89.7)

[2] 有川,他,「テキストデータベース管理システムSIGMAとその利用」,情処情報学基礎14-7,(89.7)

[3] 鶴林,他,「文書検索システム「検蔵君」」,第43回情処全大,6-61,(91.10)

[4] 山田,他,「１２０万トランジスタ辞書検索プロセッサ(ＤＩＳＰ)」,信学会春季全大C-660,1990

[5] Kato,et al."An Index-Free Full Text Search Machine for Large Japanese Text Base," proc. of Advanced Database System Symposium '89

[6] 畠山,他,「自由語検索のための同義語異表記展開方式」,第39回情処全大 2N-7 (89.10)

[7] 川口,他,「自由語検索のための高速検索方式」,第39回情処全大 2N-8 (89.10)