フルテキストサーチシステム Bibliotheca/TSの開発(1)
―システムの概要―

2C−3

浅川悟志　川下靖司　坂田　淳　畠山　敦
(株)日立製作所

1．はじめに

近年、ワードプロセッサを始めＯＡ機器の普及拡大に伴い、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"Bibliotheca/TS"を開発したので、報告する。

2．システムの概要

2.1　構　成

Bibliotheca/TSは、ＴＣＰ／ＩＰプロトコルを用いたクライアント・サーバシステムである。図１に構成を示す。

（１）BIBLIOTHECA/TS-CL(Text Search - Client)

サーバに対して検索要求を出すためのクライアントソフトウェアである。ＧＵＩにはＭｏｔｉｆを採用している。検索条件入力、結果一覧表示、テキストハイライト表示、履歴保持などの機能を有する。また、ＡＰＩとして、Ｃのライブラリ関数群を提供するようにした。

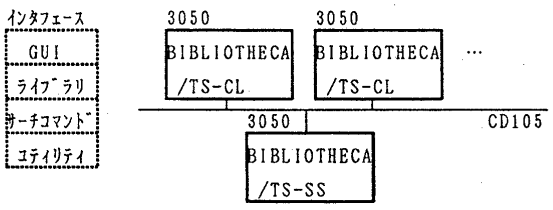

図１　Bibliotheca/TS構成概要

（２）BIBLIOTHECA/TS-SS(TS - Software Server)

クライアントからの検索要求（サーチコマンド）を処理するためのサーバソフトウェアである。同時に複数台のクライアントと通信セッションを確立できるようにした。また、ＤＢ構築やメンテナンスを行うためのユティリティ機能も持たせるようにした。

2.2　特　長

Bibliotheca/TSは、以下の特長を持つ。

（１）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方式である。スキャンの必要性を判定するために、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た。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を採用した。

この階層プリサーチ方式を用いた検索処理の流れは、図２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。そして最後に、必要に応じてテキストデータをスキャンし、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。

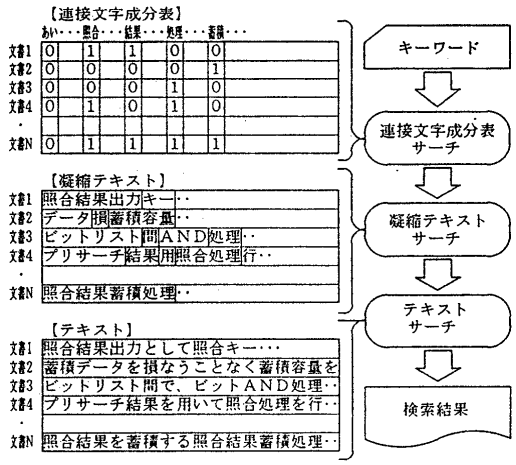

図２　階層プリサーチ方式

Development of Full Text Search System Bibliotheca/TS (1)　Outline
Satoshi ASAKAWA, Yasushi KAWASHIMO, Jun SAKATA, Atushi HATAKEYAMA
HITACHI, Ltd.

（２）サーチエンジンエミュレータ

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た。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た。

（３）同義語・異表記展開処理

ユーザの指定した検索タームと同義なタームを、同義語辞書を用いて展開するようにした。また、外来語などのカタカナ表記のゆれは、部分文字列の置換規則により、異表記を生成するようにした。これらを一括して探索することにより、フルテキストサーチの課題である検索もれの解消を図った。

（４）複合条件判定処理

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検索する機能を持たせた。この機能により、きめの細かな検索が行えるため、的確な目的の文書を捜し出すことができる。

３．性能評価

階層ブリサーチの傾向として、文字成分表での絞り込み率や、検索条件の特性により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、複合文字種の場合にはテキストデータのサーチを行う必要が生じる。

これらの要因から、以下４種類の条件下で検索処理時間を測定した。

（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する場合は、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する場合は、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、複合条件判定プログラムにかかる負荷が大きい。

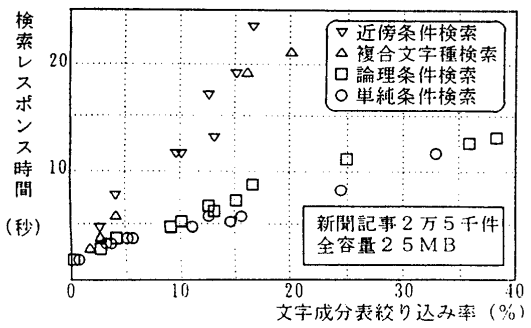

図３　システム検索性能

（３）複合文字種

漢字とカタカナの組合せなど、異なる文字種での単純条件検索の場合は、条件式は（１）と同じだが、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する場合は、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。

以上の条件での測定結果を図３に示す。検索時間は、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。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秒であることが分かる。これは、通信と条件式の解析およびサーバ内部でのコマンド実行に伴うオーバヘッドである。

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。複合文字種と近傍条件及び単純条件と論理条件の間で顕著な差がみられるが、これは複合条件判定処理にかかる負荷に比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。

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、ＤＢの検索処理速度は、１．５ＭＢ／ｓ～１２ＭＢ／ｓであるといえる。

４．おわりに

今回開発したBibliotheca/TSは、以下の特長を持つ。

（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

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」，情報学基礎14-6（89.7）