情報検索システム AIR の実現

4Y－1

林 雅樹† 江口 毅† 酒匂 直人‡ 竹田 正幸† 松尾 文碩†
†九州大学大学院システム情報科学研究科 ‡九州大学工学部

1. まえがき

現在，九州大学大型計算機センターでは，AIR とよばれる情報検索システムによって INSPEC 検索サービスが行なっている．AIR は富士通の IBM 互換の OS のもとで動作し，大部分は Fortran で書かれている．しかし，Fortran コンパイラの改版に伴い，再コンパイルができなくなっており，将来のサービス継続が危ぶまれている．そこで，著者らが開発した Seep によって AIR を UNIX システム上に再構築することにした．

INSPEC データベースは大規模であり，最大の INSPEC-A(物理学) では文献数が約 350 万に達する．そのため，直接データ転送が可能な VFL-FS ファイルシステムを用いた入出力や，効率的な文書ファイル，文書参照ファイルの圧縮を行なうことになる．

本論文では，それらを用いた効率的な AIR の構成法を述べる．

2. VFL-FS ファイルシステム

UNIX においては，通常，ファイルは UFS と呼ばれるファイルシステムによって管理されている．これは，小さなファイルを効率的に管理するためのファイルシステムであり，AIR のような大規模情報検索システムには適していない．

そのため，VFL-FS とよばれるファイルシステム上に AIR を構築した．VFL-FS は連続領域割り当てと直接転送という二つの特徴をもったファイルシステムである．この二つの特徴は大きなファイルの入出力を行なうのに非常に適している．

VFL-FS ファイルシステムを構築する際にファイルの領域確保を行なう際にブロックサイズを指定する必要がある．VFL-FS はこのブロックサイズの値の倍数でのみファイル領域の確保を行なうことから，効率的な領域確保という面からみれば，できるだけブロックサイズは小さい方がよい．このブロックサイズの値を数種とってみて，同サイズのファイルを読み込む調査を行なったところ，ディスク媒体のトラックサイズ (40960Byte) をブロックサイズとしたときが最も速いことがわかった．

3. データ構造

本システムのデータ構造は図 1に示す通りである．

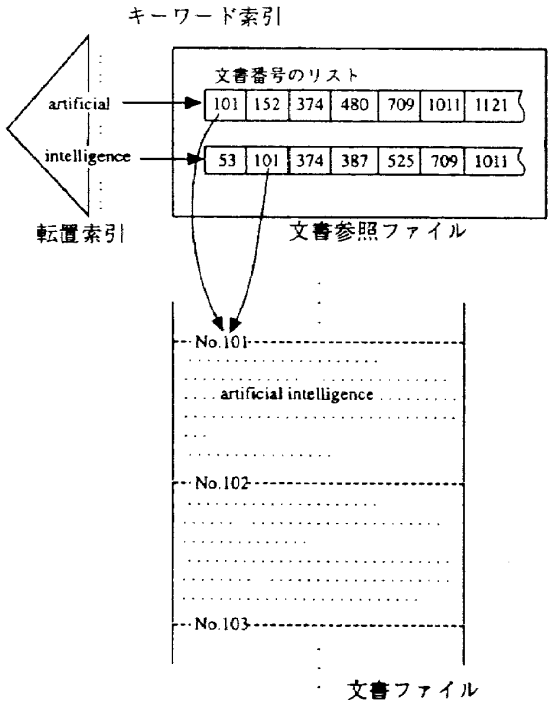

図 1 データ構造

索引部には，B^+ 木を採用し，さらに高頻度語については，主記憶上の内部索引により参照できるようにする．これにより，頻繁に検索が行なわれる高頻度キーワードについての検索が，高速化される．

それぞれのキーワードからはポインタにより，文書参照ファイル中のそのキーワードの含まれる文書の番号リストを参照できる．

Realization of High Peformance Information Retrieval System

Masaki Hayashi†, Takeshi Eguchi†, Naoto Sakou‡, Masayuki Takeda†, and Fumihiro Matsuo†
†Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, Japan
‡Faculty of Engineering, Kyushu University, Fukuoka, Japan

4. 文書ファイルの圧縮

文献検索システムの文書は通常，一文字あたり 8 ビットの符号で表現されており，かなり冗長である．文字や単語をその生起頻度に応じて可変長符号で表すことにより，冗長度を減らすことができる．

この方法ではハフマン符号が最適であることが証明されているが，文字を圧縮単位としたハフマン符号による英文の圧縮では，1/2 倍程度の圧縮効果しかないことが知られており，さらに圧縮効果を上げるには，単語や N-gram などの文字列を単位とした圧縮方法をとらざるをえない．しかし，ハフマン符号は単語を圧縮単位とした場合のように要素数が非常に大きい場合は，処理が面倒で符号化および復号化に要する計算量が大きくなる．

そこで，AIR では著者らが開発した QOC(Quasi-Optimum Code) と呼ぶテキスト圧縮技法を用いる．QOC は高頻度単語に対しては，単語を圧縮要素とする順位符号 (Rank Code) を用い，低頻度単語に対しては，文字を要素とする圧縮を行なう．順位符号はハフマン符号のように最適ではないが，Zipf の法則 [1] が成立する言語に対しては，その圧縮率はハフマン符号より 2.7% 悪いだけである [2]．一方，符号化と復号化に要する計算量において，順位符号はハフマン符号に決定的に勝っている．

QOC により INSPEC テープの圧縮を試みたところ，抄録や標題のみならず，誌名，著者の所属機関名などの書誌情報に対しても，この技法が有効であり，これらの項目については約 1/4 にデータを圧縮することができた [2]．

5. 文書参照ファイル

AIR では，文書内のキーワードの生起位置情報を持たない単純転置索引を用いる．ここでは，この単純転置索引の効率的な構成法を述べる．単純転置索引を用いた情報検索システムには，リストの長さの分布に著しい偏りがあるという特異性がある．つまり，高頻度キーワードの生起回数は扱う総文書数に比例して増加し，それについての文書番号リストの長さは数十万にもなるのに対し，低頻度キーワードにおいては，文書中における全出現単語異なり数のうち，約半数の単語についてのリストの長さは 1 ということである．これらの極端なリストを混在させて一様に管理すると，二次記憶領域の非効率的確保という問題が発生する．そのため，頻度によって異なる方法によって管理する．

しかしながら，高頻度語や低頻度語をどこで分けるかという問題が残る．この問題についてはプロトタイプを試験運用しながら決めていく．

5.1 高頻度キーワード文書参照ファイル

高頻度キーワードの文献番号リストはリストの長さが非常に長く，各番号間の差分値が非常に小さい．そこで，各番号の差分値リストを二次記憶上に格納する．これ自体は圧縮になっていないが，差分値リストは圧縮に適したものになっている．したがって，この差分値リストを順位符号を用いて圧縮したものを二次記憶上に格納する．ここで順位符号を用いたのは，符号化および復号化に要する計算量が最適符号であるハフマン符号に比べて少ない上，圧縮率も悪くないからである [3]．

5.2 低頻度キーワード文書参照ファイル

低頻度キーワードの文献番号リストは高頻度キーワードの文献番号リストのようにそれぞれのリストの増加分が予想できない．しかし，生起回数毎にまとめて管理することにより，その文献追加時のリストの増分が予想できるようになる [4]．そこで本システムでは，生起回数 1 から m までの単語を生起回数毎にまとめて管理する方法を採用した．これにより，リストの増分の予想が可能になり，効率的な二次記憶領域の管理が可能になる．

5.3 中頻度キーワードの文書参照ファイル

低頻度キーワードにも高頻度キーワードにも入らない中頻度語の文献番号リストは，論理ブロックを用いて管理する [5]．論理ブロックは一つの物理ブロックを複数の管理単位に分割したものである．この論理ブロックの大きさの異なるものを数種類用意し，低頻度なものほど小さな論理ブロックを割り当てるようにすれば，無駄領域を小さくできる．この論理ブロックの数やサイズを適当な値に設定することにより，効率的な管理法が実現できる．

6. むすび

以上に述べた方法により，今回 AIR のプロトタイプを作成した．これにより，これらの方法の総合的な処理速度，効率などを評価し，公開に向けての問題点等を検討する．

参考文献

1) Zipf, G.K.: *Human Behaviour and the Principle of Least Effort*, Addison-Wedley, Combridge, Mass. (1949).

2) 松尾文碩，二村祥一: 順位符号に基づく英文二次文献情報のデータ圧縮法，情報処理学会論文誌，第 28 巻，第 3 号 (1987).

3) 林，小出，竹田，松尾: 情報検索システムにおける高頻度キーワードの文書参照ファイルの圧縮について，第 55 回情報処理学会全国大会講演論文集 (1997).

4) 松尾，佐藤，高山: 情報検索システムにおける文書参照ファイルの効率的構成 **36**(6), 1486–1494 (1995).

5) 小出，林，竹田，松尾: 情報検索システムにおける文書参照ファイル，第 54 回情報処理学会全国大会講演論文集 (1997).