高速追加処理を可能とする転置ファイル構成法

1H-8

篠原 武, 二村 祥一, 松尾 文碩
(九州大学大型計算機センター)

1. はじめに

九州大学大型計算機センターでは,筆者らが設計・開発した情報検索システムAIR[1]を用いて,INSPECなどの文献検索サービスを行っている[2]. AIRは検索速度や更新処理速度,ディスク使用効率のすべてにおいて,他の同種のシステムより圧倒的にすぐれている.

AIRでは,キーワードによる検索のために転置ファイルを用いている.文献検索システムにおける更新処理のほとんどはデータの一括追加によるものであるため,AIRでは,転置ファイルの更新を文献の追加に伴うものに限定し,処理を高速化している.しかし,追加を繰り返すと徐々に処理速度が低下するという問題があり,そのため転置ファイルを定期的に再構成する必要があった.

そこで本稿では,キーワード転置ファイルへの追加を高速化し,上の問題を解決する方法について議論する.ここで提案する新しい転置ファイル構成法では,キーワードが生起する文献の間隔が非常に偏った分布をすることを利用した,単純かつ効率的な転置ファイルの圧縮符号[3]を採用し,さらに,キーワードを頻度によって分類し,各類毎に追加によって更新を受ける領域を辞書式順序に配置する.この方法によると,ディスクの使用効率も良く,追加処理を非常に高速に行うことができ,しかも追加を繰り返してもこれらの性能は劣化しない.もちろん,検索も高速に行える.

2. キーワード転置ファイルの構造

AIRのキーワード転置ファイルは,図1に示したように,索引ファイルと文書参照ファイルから構成される.

索引ファイルには,キーワードに対する文書参照ファイル中のデータへのポインタが格納される.索引ファイルは,B木などを用いて実現されるが,これについては,筆者らはすでに,キーワードの生起頻度が偏っていることを利用した高速単語索引[4]の技法を確立している.

文書参照ファイルには,各キーワードに対して,そのキーワードが現れている文書の参照番号リストが格納されている.検索や追加などの処理効率に与える影響は,索引ファイルよりも文書参照ファイルの構成の方が大きい.そこで,本稿では文書参照ファイルの構成法を中心に議論する.

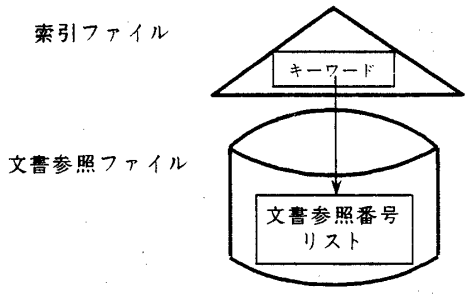

図1. キーワード転置ファイルの構造

キーワード転置ファイルの設計において注意すべきことは,次の3点である:

(1) 各キーワードに対する文書参照番号リストへのアクセス速度.

(2) 文書参照ファイル全体でのディスク使用効率.

(3) 更新処理の速度.

ただし,更新処理は文献データの追加に伴って行うものだけを考える.

ディスク装置とのアクセスはブロック単位で行われる.(1)は文書参照番号リストを出来るだけ少ないブロックに格納すればよい.(2)は出来るだけ詰め合わせて文書参照番号リストを格納すればよいが,更新処理を行うと,ファイルの再構成をしないと文書参照番号リストが寸断されてしまう.そこで,(1)と(2)をある程度両立させ,(3)を犠牲にしないために,文書参照番号リストの末尾に適当な大きさの追加用空領域を設ける方法が考えられる.しかし,この方法では不十分である.

3. 転置ファイルの追加処理

文書参照番号リストの末尾に適当な大きさの追加用空領域を設けておくと,転置ファイルを構築した直後の追加処理は,各文書参照番号リストの末尾の空領域にデータを追加するだけでよいので,高速に行える.しかし,追加を繰り返して空領域がなくなった場合に,新たに確保したディスク領域を連結していくと,文書参照番号リストは徐々に寸断され,検索処理速度が低下し,しかも更新処理を受ける領域が散在するので,追加処理速度も低下する.これが,はじめに述べた問題の主な原因である.

キーワード転置ファイルの更新は,文書毎に処理するよりも,追加された文書を一括して処理する方

Organization of Inverted Files for Rapid Addition
Takeshi SHINOHARA, Shouichi FUTAMURA, Fumihiro MATSUO
Kyushu University

が効率が良い．一括処理の場合には，キーワードの辞書式順序で文書参照番号リストを更新するので，追加によって書き換えられるディスクの領域がキーワード順に並んでいると，1回のブロックへのアクセスで複数のキーワードの処理が行え，しかも同一ブロックを2回以上アクセスする必要がない．転置ファイルの構築時には，追加のための空領域をキーワード順に並べることは容易であるが，素朴な方法では，追加を繰り返すとその順序が乱れ，処理速度が低下する．

4. キーワードの生起頻度を考慮した構成法

文書データにおけるキーワードの生起頻度は非常に偏っている．頻度の高いキーワードは，個数は少ないが，追加される文書数にほぼ比例して生起頻度が大きくなる．それに対し，頻度の低いキーワードの個数は非常に多いが，文書が追加されても個々のキーワードの生起頻度が増加するとは限らない．しかし，低頻度のキーワード全体では，頻度が増加するものの個数は少なくない．このように，高頻度のキーワードと低頻度のキーワードの生起特性は大きく異なるので，文書参照番号リストの管理は高頻度のものと低頻度のものに分けて行う必要がある．すべてのキーワードを一様に管理すると，高頻度キーワードの文書参照番号リストが低頻度のものによって寸断され，検索処理速度の低下を招く．また，追加を繰り返した場合に追加のための空領域の順序が乱れるのも，低頻度キーワードの影響が大きいと考えられる．

4.1 高頻度キーワード

高頻度キーワードの文書参照番号リストの追加用領域の大きさは，キーワードの頻度と文献追加の割合によって決定しなければならない．追加用領域が大き過ぎると1回のディスクブロックの読み書きで処理できるキーワード数が減少し，逆に小さ過ぎると領域が不足し領域を追加する回数が増加する．

追用領域を格納するブロックには追加用領域だけを格納し，キーワード順に格納するようにする．このようにすれば，確実に1回のディスクブロックの更新によって複数のキーワードの処理ができる．追加用領域が不足すると新たにディスク領域を確保し連結しなければならないが，図2に示すように，新しい領域を追加用領域の後ろに連結せずに直前に連結しデータを移動すれば，追加用領域はもとの位置に置くことができる．

また，連結を繰り返すと文書参照番号リストを格納するのに使用されるブロック数が多くなり，検索処理速度が低下するので，これを防止するために，連結を数回行う毎に領域の詰め合わせを行い，追加用領域の大きさも変更する．もちろん，新しく確保した追加用領域はキーワード順に並ぶように配置しなければならない．このために，追加用領域の大きさによってキーワードをグループ化し，各グループ毎に追加用領域をキーワード順に格納する．

高頻度のキーワードが出現する文献の間隔(gap)は小さいものが多いので，その文書参照番号リストはGap Code[3]によって効率よく圧縮できる．

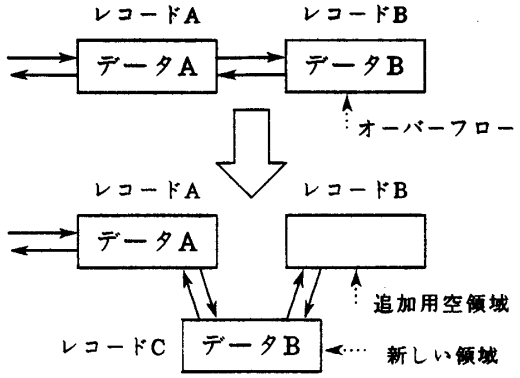

図2. 領域の追加

4.2 低頻度キーワード

低頻度キーワードは，頻度によってグループ化し，高頻度キーワードの追加用領域と同様にキーワード順に並べておく．低頻度のキーワードが追加によって頻度が増加し領域が不足した場合には，単にもとの領域を解放し別の頻度グループに領域を移動する．頻度1および2のキーワードは，全キーワード数の約60%を占めるが，これらについては，索引ファイルに文書参照番号リストへのポインタの代わりに直接文書参照番号を格納する．もちろん，追加を繰り返すと低頻度から高頻度へ移動する場合もある．

低頻度キーワードの文書参照番号リストに対しては，Gap Codeの圧縮効率が低いので圧縮は行わない．

5. おわりに

以上で説明した転置ファイルの構成法に基づいてAIRの改訂作業を急いでいる．現在までの実験で，新しいキーワード転置ファイルでは，ディスクの使用量はこれまでの55%程度になり，追加処理時間は35%以上短縮でき，追加を繰り返しても追加処理速度に悪影響を及ぼさないことは確認できた．追加処理は，バッファの割当てなどの最適化によって，さらに高速化できると考えている．検索処理効率の向上については，文書参照番号リストは頻度が高いキーワードのものほど圧縮率が大きくなるので，単純な質問に対しても平均的には，3倍程度の高速化は得られるであろう．また，キーワードの前方一致による質問処理は，文献検索などにおいて重要であるが，非常に多くの文書参照番号リストへのアクセスを必要とする．その大部分は低頻度のものであり，それらは索引ファイルや低頻度用の領域にキーワードの辞書式順序で配置されるので，前方一致による質問処理に対しては，より顕著な効率向上が期待できる．

参考文献

[1] Matsuo, F., Futamura, S., Shinohara, T.: Efficient Storage and Retrieval of Very Large Document Databases, Proceedings of the Second International COnference on Data Engineering, (1986), 456-463.

[2] 二村，篠原，松尾：情報検索システムAIRによるINSPECの検索，九州大学大型計算機センター広報，Vol.17, No.1, (1984), 1-22.

[3] 篠原，松尾，二村：高速ブール演算のための効率的転置ファイル構成法，情報処理学会第31回全国大会講演論文集，7B-9 (1985).

[4] 松尾，二村，篠原：高速単語索引，同上26回大会，7F-4 (1983).