6N－2

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

林 雅樹　小出 東洋　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

1. まえがき

情報検索システムのファイルは，通常，文書そのものを格納した文書ファイルと，文書に対する索引部である転置ファイルとによって構成される．転置索引ファイルは索引と文書参照ファイルからなる．文書参照ファイルは，索引の見出し語であるキーワードがもつ文書番号等の線形リストを格納したファイルである．文書参照ファイルでは，線形リスト長の分布に著しい偏りがある．高頻度キーワードの線形リストは格納文書数に比例して増大し，大規模システムでは数十万以上になる．このリストをそのまま二次記憶に格納すると，膨大な領域を必要とするばかりか，検索速度の低下にもつながる．本稿では，この高頻度キーワードの文書参照ファイルの圧縮について述べる．

2. 順位符号

情報検索システムにおいて，文書参照ファイル中のデータを圧縮する目的は，大きく分けて次の２つがある．

(1) 二次記憶上に要するデータ領域を縮小する．

(2) システム全体の処理時間に大きく影響するディスクアクセスタイムを減少する．

元来，データ圧縮技法として，ハフマン符号が最適（コンパクト）符号として知られている．しかし，ハフマン符号は最適符号化において圧縮要素すべての生起確率順位を必要とする．しかし，リストへのデータ追加によって，そのリスト中の要素数の変化はもちろん，その生起確率分布までが変化することが考えられる．そうなると，リスト全体の再通読・再符号化が必要となるため，オンラインシステムである情報検索システムにおいては，ハフマン符号を用いて常に最適なデータの圧縮形態を維持するには処理が複雑になるばかりか高速性に不満が生じる．ゆえに，本研究では文献番号リストの圧縮にハフマン符号を適用することは困難と考え，順位符号による圧縮を試みる．

順位符号 (Rank Code) とは，圧縮要素の生起頻度の順位そのものを可変長２進数によって表現した符号である．順位 $r(\geq 1)$ の要素 e の符号を ε とすると，ε はビット長 $\lfloor \log_2 r \rfloor$ の２進数である．これをマトリックス (matrix) と呼ぶ．しかし，このままでは復号ができないため，マトリックスの前にプレフィックス (prefix) と呼ぶ固定長部を置き，ここにマトリックスのビット長を２進数で表したものを入れる．このプレフィックスとマトリックスからなる符号が順位符号である．順位符号のマトリックスにおいて，最上位桁を除去するのは，この桁が常に１であるため，これがなくても復号が可能であるためである．表 1 に順位符号による符号化の例を示す．この例では prefix の長さを 4 bit にしてある．

表 1 順位符号による符号化

rank	prefix	matrix	rank code
1	0000	-	0000
10	0001	0	00010
11	0001	1	00011
100	0010	00	001000
101	0010	01	001001

Compression of Document Reference File for High Frequency Keywords in Information Retrieval System
Masaki Hayashi, Haruhiro Koide, Masayuki Takeda, and Fumihiro Matsuo
Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka, 812-81 Japan

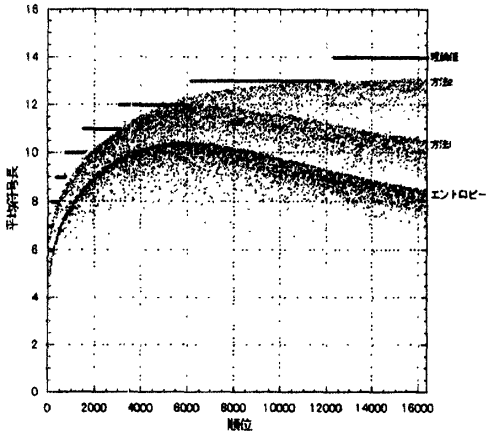

図 1 平均符号長

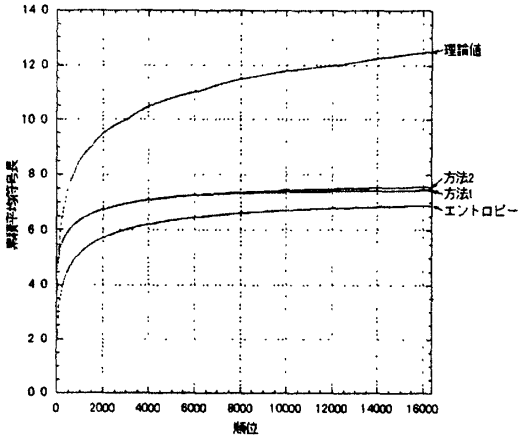

図 2 累積平均符号長

3. 文書参照ファイルの圧縮

文書参照ファイルを圧縮する場合，単純に文献番号リストを圧縮するのではなく，文献番号の差分値リストを圧縮する．しかし，これをそのまま生起頻度順に順位をつけて圧縮したのでは，ハフマン符号の場合と同様に，リストの更新の際に再通読・再符号化が必要となるためそのままでは適用し難い．そこで，対象となる文書参照ファイルは高頻度キーワードに対するものであり差分値は小さいと考えられるため，差分値をそのまま順位符号における順位として圧縮することを考えた．

高頻度語については Zipf の法則 [1] が成立し，順位 r の語の生起確率を $p(r)$ とすると，$p(r)$ は次の式で表される．

$$p(r) = 0.1/r$$

そこで，1 文献中の単語数を w とすると，順位 r の語は 1 文献中に $0.1w/r$ 回生起するから，$r/0.1w$ を順位 r の語の文献番号リストの差分値の平均と考えた．これより順位 r の語の文書参照ファイルを順位符号を用いて圧縮した際の平均符号長 l_r，順位 1 から順位 r までの語の累積平均符号長 S_r はそれぞれ，

$$l_r = 4 + \left\lfloor \log_2 \frac{r}{0.1w} \right\rfloor$$

$$S_r = 4 + \frac{1}{n} \sum_{k=1}^{n} \left(\left\lfloor \log_2 \frac{k}{0.1w} \right\rfloor \right)$$

と表せる．

そこで，INSPEC テープ [2] 25 年分のデータから，それぞれの順位の語に対する文献番号リストを，差分値の生起確率の順位を順位符号の順位とした圧縮法 (方法 1)，差分値を順位符号の順位とした圧縮法 (方法 2)，エントロピー，Zipf の法則から求めた平均符号長のグラフを図 1 に示す．図 2 はさらに，これらを順位 1 からの累積で示したものである．

グラフでは理論値と実際の値とは離れているが，これは Zipf の法則がごく高頻度 (順位 100 くらいまで) の語でしか成り立たないからである [3]．

図 2 より，圧縮する順位の下限を大きくすれば大きくするほど圧縮効率は良くなることがわかる．

4. むすび

本稿では情報検索システムの文書参照ファイルを圧縮する際の効率的な圧縮法について述べた．さらに，どれくらいの生起回数の語までを高頻度語として圧縮するのが効率がいいかについて調べた．これらのことをふまえて，新たな情報検索システムを構成することが今後の課題である．

参考文献

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

2) Aithison, T.M., Martin, M.D. and Smith, J.R.: Developments towards a Computer Based Information Services in Phisics, Electrotechnology and Control, *Inform. Strage and Retrieval*, 4(2), 177–186 (1968).

3) Fumihiro, M.: On word occurrence in scientific and technological texts, 情報処理学会自然言語処理研究会資料,46-2(1984).