

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

松 尾 文 碩[†] 佐 藤 誉 夫^{††} 高 山 悟^{†††}

情報検索システムのキーワード転置ファイルには、索引の見出し語（キーワード）の内容を格納したファイルがあり、これを文書参照ファイルということにする。見出し語の内容は、文書番号あるいは見出し語の生起位置の線形リストであり、この長さの分布は非常に偏っている。文書参照ファイルにおいて、長短リストを同形式で記憶し、2次記憶アクセス回数を減らすためにブロックサイズを大きくすると、低頻度キーワードのために非常に大きな無駄領域が生じる。本稿では、英文科学技術抄録文に関して、個々の低頻度キーワードの増加は予測できないが、生起回数が同一なものをまとめると、群として増加が予測できることを利用して、低頻度キーワードリストを生起回数ごとに群として管理する方法を提案した。この方法によって無駄領域を大きく減少させることが可能である。

Efficient Organization of Document Reference File in Information Retrieval System

FUMIHIRO MATSUO,[†] TAKAO SATOU^{††} and SATORU TAKAYAMA^{†††}

Keyword inverted file of information retrieval system consists of the index and its content file called document reference file here. These contents are linear lists of the document numbers or the positions of the keyword occurrences, so the distribution of the linear list lengths is very skew. Storing the lists in the same form without regard to their lengths yields very large useless area for the low frequency keywords, when the block size is set large to decrease the secondary memory access times. As for the low frequency English keywords in scientific and technical documents, the increase of total occurrences of the words occurred n times with increasing the stored documents can be estimated, though the increase of each word cannot be done. By using this estimation, this paper proposed a method that stores the low frequency keyword lists as groups in which the lists of the keywords occurring same times are placed. This method can reduce the useless area sharply.

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 (document file) と、文書に対する索引部である転置ファイル (inverted file) とによって構成される。転置ファイルには、キーワードに関するものと、著者、雑誌名、出版社などの書誌的情報に関するものがある。キーワードに関する転置ファイルをキーワード転置ファイルと呼び、それぞれの書誌的情報に関するものを属性値転置ファイルと呼ぶことにする。ここで属性値転置ファイルというのは、

そのファイルの特性が関係型データベースシステムにおける属性 (attribute) についての索引部と本質的に同じであるという認識による。一方、キーワード転置ファイルは情報検索システム固有の特性をもち、ファイルサイズが属性値転置ファイルよりずっと大きい。キーワード転置ファイルは、情報検索システムの性能に最も関係しているため、その構成法は重要である。

キーワード転置ファイルの特性は、言語現象を反映しているため、文書で使用される言語によってその特性が異なる。英語の場合は、キーワードは検索時の事後結合句 (post coordinated clause) 作成のため、単一語 (single word) が選ばれる。日本語の場合、キーワードの選択について英語のように定説があるわけではない。本稿では英文、特に科学技術抄録文に対するキーワード転置ファイルの構成について論じる。本稿では、属性値転置ファイルについては論じないので、以下、転置ファイルとはキーワード転置ファイルのみを

[†] 九州大学工学部

Faculty of Engineering, Kyushu University

^{††} 日立超 LSI エンジニアリング(株)

Hitachi ULSI Engineering Corp.

^{†††} 九州大学大学院工学研究科

Division of Engineering, Graduate School, Kyushu University

指すことにする。

転置ファイルの索引(図1参照)に関しては、B-treeをはじめ多くの研究がある¹⁾。索引の見出し語であるキーワードがもつ内容は、文書番号等の線形リストである。しかし、この線形リスト長の分布は、文書集合における英単語の生起分布の偏りを反映して非常に偏っている。このような線形リストの集合を効率的に蓄積するための研究は皆無に近い。本稿では、この線形リスト集合を格納するファイルの効率的構成法を提唱し、その方式を評価した。

2. 転置ファイルの構成と問題点

転置ファイルは、通常、図1に示すように索引(index)と文書参照ファイル(document reference file)と呼ぶ二つのファイルから構成される。一般的に、索引はB-tree²⁾やB⁺-tree³⁾のような多岐平衡木によって実現することが好まれる。木構造索引の場合、あるキーワードに対応する葉あるいは節は、そのキーワードの内容(content)へのポインター(content pointer)をもっている。文書参照ファイルは、内容の集合であり、その意味では、内容ファイルといってもよい。

内容がそのキーワードを含む文書番号の線形リストの場合、この転置ファイルを単純転置ファイル(simple inverted file)という⁴⁾。文献番号のほかに、文書内のキーワードの生起位置をリストにもつ場合、隣接演算型転置ファイル(adjacency operation type inverted file)という⁴⁾。単純転置ファイルは、ある意味では隣接演算型転置ファイルの縮退形であると考えられる。キ

ーワードの生起頻度が高いほど縮退の度も高いが、大規模英文科学技術2次文献の情報検索システムの場合、キーワードの選び方にもよるが、各キーワードがもつ生起位置の個数は平均的にはおおよそ2であり、平均すれば1/2程度の縮退である。特に、本稿の議論の対象となる低頻度キーワードの部分に関しては、2種類の転置ファイルの線形リスト長の分布には差異がほとんどない。

図1は、単純転置ファイルを表わしていて、キーワード artificial の内容である線形リストは、文献番号101番、152番、374番などの文献中に artificial が1回以上生起していることを示している。

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生起回数は格納文書数に比例して増加し、大規模システムでは高頻度キーワードの線形リスト長が数十万以上になるのに対し、約半数のキーワードのリスト長がシステムの規模に無関係に1である。長短の線形リストを同一形式で2次記憶に蓄積した場合、低頻度キーワードのリストを記憶するために大量の無駄領域が生じる。

本稿では、この無駄領域を減少させるための一つの解決法を示す。

3. 低頻度単語の生起

英単語の生起確率については、Zipfの法則⁵⁾が有名である。この法則によれば、ある単語 w の生起頻度の順位を r とすると、 w の生起確率 $p(r)$ は、次式で表わされる⁶⁾。

$$p(r) \approx k/r, \quad (1)$$

ここで、 $k \approx 0.1$ 。

しかし、 $\sum_r 1/r$ は発散するので、 $k=0.1$ の場合、

$\sum_{r=1}^{12,367} p(r) > 1$ となり、 r が大きい単語、すなわち低頻度に関しては、(1)式は成立しない。松尾は、英国電気学会(Institution of Electrical Engineers)が刊行している抄録誌の機械可読形であるINSPECテープ⁷⁾の科学技術抄録文を調査し、Zipfの法則が成立するのは高頻度単語に関してのみであり、低頻度単語については次式が成立することを示した⁸⁾。

$$p(r) = c/r^2, \quad (2)$$

ここで、 c は定数。

(2)式が成立するならば、 $\sum_{r=1}^{\infty} 1/r^2 = \pi^2/6$ であるので、発散の問題は解消する。

INSPECテープは、物理学、電気・電子工学、制御工学、計算機科学、情報工学の分野の抄録を収録して

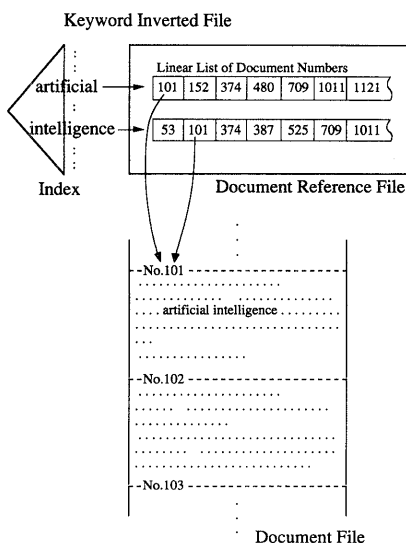


図1 主題検索のためのファイル構成

Fig.1 File organization for retrieval on subject.

表1 INSPEC テープ抄録文における I_n/D
Table 1 I_n/D in abstracts of INSPEC-tapes.

n	Observ. in Abstracts of INSPEC			Theory	
	A	B	C	Eq. (4)	Booth ⁹⁾
1	.460076	.453307	.440893	.422650	.5
2	.135999	.124616	.137833	.130137	.166667
3	.069391	.067328	.071420	.069249	.083334
4	.042471	.041879	.043618	.044631	.05
5	.030071	.028976	.030453	.031822	.033333
6	.022394	.021922	.022513	.024161	.023810
7	.017313	.017182	.017515	.019151	.017857
8	.014049	.013421	.013924	.015663	.013889
9	.011383	.011197	.011804	.013120	.011111
10	.009515	.009881	.010521	.011198	.009090
20	.003370	.003294	.003395	.003954	.002381
30	.001765	.001822	.001721	.002152	.001075
40	.001218	.001193	.001306	.001398	.000610
50	.000821	.000901	.000892	.001000	.000392
100	.000281	.000305	.000313	.000354	.000099

いる。これを、物理学、電気・電子工学、それら以外の3分野に分け、3分野の抄録文集合をそれぞれA, B, Cと呼ぶことにする。約20%の抄録文が二つ以上の集合に属している。この3分野のいずれでも(2)式が成立する⁹⁾。このことから他の分野の科学技術抄録文についても(2)式が成立するものと考えられる。

いま、単語をつづりだけで区別し、つづりが異なる文字列は異なった単語とみなす。ある文書集合における異なり単語数を D 、延べ単語数を T 、 n 回生起する異なり単語数を I_n とすると、(2)式から次の関係が導かれる⁹⁾。

$$D = \sqrt{2cT}. \quad (3)$$

$$\frac{I_n}{D} = \frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}}. \quad (4)$$

(3), (4) 式の導出は、付録1に示す。(3)式については、INSPEC テープの抄録文における実測値も $D \propto \sqrt{T}$ であることを示している⁹⁾。表1に、INSPEC テープの抄録文における I_n/D の実測値と(4)式およびBoothの式⁹⁾ ($I_n/D = 1/n(n-1)$) の計算値を示した。表1のA, B, Cは、10年分のINSPEC テープの抄録文を、前に述べた3分野に分けた文献集合に関する値である。A, B, Cにおける T は、それぞれ87,354,577, 37,606,323, 23,391,467であり、 D はそれぞれ274,185, 154,231, 127,836である。表1からわかるように(4)式はBoothの式より近似度が高い。

データ量の増加に伴う I_n の増加に関しては、 D の増分 ΔD に対する I_n の増分 ΔI_n は、(4)式から次のようになる。

表2 INSPEC テープ抄録文における ΔI_n
Table 2 ΔI_n in abstracts of INSPEC-tapes.

n	A		B		C	
	Observ.	Eq. (5)	Observ.	Eq. (5)	Observ.	Eq. (5)
1	9,934	9,114.0	6,146	5,516.0	5,274	4,844.8
2	2,786	2,806.3	1,840	1,698.4	1,556	1,491.4
3	1,729	1,493.3	974	903.8	975	793.8
4	1,005	962.4	533	582.5	594	511.6
5	637	686.2	343	415.3	375	364.8
6	435	521.0	327	315.3	229	277.0
7	376	413.0	289	249.9	153	219.5
8	403	337.8	121	204.4	95	179.5
9	237	292.9	87	171.2	138	150.4
10	85	241.5	150	146.1	163	128.4

$$\Delta I_n = \left(\frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}} \right) \Delta D. \quad (5)$$

表2に示すように、(5)式もこの種の推定式としては近似度が高い。表2では、A, B, Cの T は、それぞれ表1のものと同一であり、 ΔT はそれぞれ11,697,810, 5,114,078, 3,459,838であった。(3)式から、 T の増分 ΔT に対する ΔD は、 $\Delta D = \sqrt{\frac{c}{2T}} \Delta T$ となるので、(5)式はまた、次式のように表わすことができる。

$$\Delta I_n = \sqrt{\frac{c}{2T}} \left(\frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}} \right) \Delta T. \quad (6)$$

T と ΔT は、それぞれ文書量、追加文書量に対応している既知である。したがって、(4)式と(6)式から I_n と ΔI_n を推定することができる。すなわち、個々の低頻度単語 w の $\Delta w / \Delta T$ を予測することはできないが、 n 回生起する単語を集合化すると、その集合の増加量は予測できるのである。

4. 低頻度キーワードのリスト管理

大規模実用情報検索システムでは、キーワードは不要語除去法によって選ばれる¹⁰⁾。不要語除去法とは、文書に現れる語のうち、文書の内容を同定する力のない不要語(stop word)と呼ばれる語を除いたものをキーワードとする。不要語は、通常、高頻度単語の中から選ばれるので、低頻度単語はすべてキーワードになる。そこで、本稿では低頻度キーワードと低頻度単語を同一視する。

4.1 無駄領域

2次記憶上に直接アクセスファイル(direct access file)を構成するためには、そのファイルの2次記憶のブロックは固定長でなければならない。大規模実用システムでは、文書参照ファイルは高速化のために直接

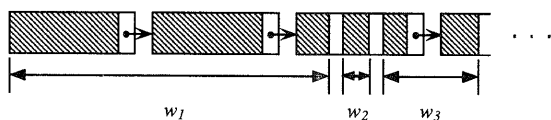


図2 非現実的線形リスト格納法
Fig. 2 Impractical storage of linear lists.

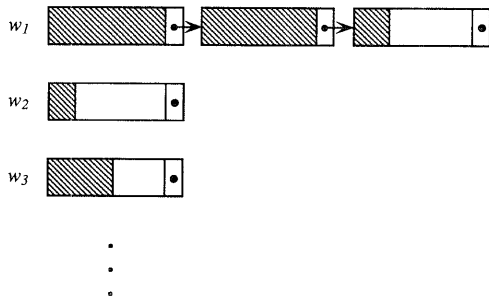


図3 線形リスト格納法
Fig. 3 Storage of linear lists.

アクセスファイル編成が採用されるからである。このとき、高頻度キーワードと低頻度キーワードのリストを図2のように混在させる構成は、2次記憶の管理が複雑になり、特に大規模実用システムでは採用が困難である。なぜなら、高頻度キーワードの生起回数は、その生起確率が(1)式に従うならば、延べ単語数 T に比例することになるので、必要な追加領域が予測できるのに対し、 w_2 のような低頻度キーワードはそのリストの増加量が予測できず、各リストの後方に適切な追加領域を確保できないためである。したがって、この構成法をとった場合、追加領域を使い切ったときにはリストの移動が必要である。また、この方式ではリストの先頭位置はブロック番地とブロック内番地によって決まる。このことは、リスト移動のための空き領域管理や索引の内容ポインターの更新手続きを煩雑にする。

そこで、大規模実用システムでは、図3のように各単語の線形リストを必ずブロックの先頭から置く構成法が好まれる。このブロックは、物理的なアクセス単位である物理ブロックではなく、物理ブロックを分割した固定長の論理ブロックであるかもしれない。しかし、ブロック内領域量の解析には、物理ブロックであるか論理ブロックであるかは関係しない。

いま、1ブロックに収容できるリストの長さを k とする。例えば、長さ1の単語についていえば、このブロックには $k-1$ の空きがある。この空きはいつ埋められるか予測がつかず、場合によっては空きの状態は変化しないかもしれない。つまり、低頻度キーワードの線形リストはほとんど使われることのない‘死んだ’

追加領域をもつことになる。

図3の構成において、1ブロックに入るリストの長さを k 、低頻度キーワードの線形リスト長を j であるとする。1ブロックのなかで、 $k-j$ の領域がすべて無駄というわけではないが、簡単のため、これをそのブロックの無駄領域と考えることにする。すると、すべてのブロックの無駄領域の大きさ V は、

$$V = \sum_{j=1}^{k-1} (k-j) I_j \\ = \sqrt{2cT} \left(k - \frac{1}{\sqrt{2k-1}} - \sum_{j=1}^{k-1} \frac{1}{\sqrt{2j-1}} \right). \quad (7)$$

(7) 式では低頻度単語の生起回数を計算しているの、隣接演算型転置ファイルの場合を扱っているようにもみえるが、単純転置ファイルの場合でも(7)式とほとんど差はないと考えられる。以下、二つのファイルには差はないものとして議論する。 $k \geq 3$ のとき、(7)式は次のように表わすことができる。

$$V \approx \sqrt{2cT} \left(k - \sqrt{2k-5/2} - \frac{1}{\sqrt{2k-1}} - \frac{1}{4\sqrt{2k-3}} + \alpha \right), \quad (8)$$

ここで、 $\alpha = 0.43640678 \dots$ 。

(8)式の導出については、付録2を参考にいただきたい。表1における T と D の値から $\sqrt{2c}$ を算出すると25~30である。そこで、 $\sqrt{2c} = 27$ 、 $T = 10^8$ とし、 $k = 100, 500, 1,000$ とすると、(8)式から無駄領域 V は、それぞれ $0.23T$ 、 $1.27T$ 、 $2.58T$ となる。隣接型転置ファイルの場合、総リスト長はおおよそ T であると考えられる。このことから、ブロック長を大きくすると、大規模情報検索システムでは無駄領域がいかに巨大になるかがわかる。

高頻度キーワードの線形リストを高速に読み込むためには、2次記憶アクセス回数を減らす必要があり、そのためには物理ブロック長はできるだけ大きい方がよい。しかし、大きくとった物理ブロックの先頭から低頻度キーワードの線形リストを置くことは、上述のように無駄領域が大きくなる。この無駄領域を減少させる方策としては、高頻度キーワードリストには物理ブロックを使用し、低頻度キーワードには論理ブロックを使う方法が考えられる。このとき、論理ブロックは長さの異なるものを数種類用意し、低頻度なものほど、小さい論理ブロックを割り当てるようにすれば無駄領域が小さくなる。しかし、このような方法は実現上の常識的な対処法である。本稿では線形リスト群による解決策を示す。

4.2 リスト群による管理

3章で述べたように、低頻度キーワードの線形リス

List Group 1 for words occurred once	List Group 2 for words occurred twice	...	List Group m for words occurred m times
--	---	-----	---

図4 低頻度キーワードの線形リスト格納領域

Fig. 4 Storage area for linear lists of low frequency keywords.

トに必要な領域および追加領域は、同じ生起回数のもを集合化すれば、推定可能であることから、図4に示したように、低頻度キーワードの線形リスト領域を切り離し、その内部を n 回生起語線形リスト群に分割する方式が考えられる。この方法では、無駄な追加領域が生じない。

実際のシステムでは、低頻度キーワード領域に入れるリストは、ある $m(m \leq k-1)$ を決めて、 m 回以下の生起単語のものとせざるをえない。すると、 m 回以下生起するキーワードに関する無駄領域の大きさ V_m は、

$$V_m = V - \sum_{j=1}^m (k-j) I_j$$

$$= \sqrt{2cT} \left(\frac{k-m}{\sqrt{2m+1}} - \frac{1}{\sqrt{2k-1}} - \sum_{j=m+1}^{k-1} \frac{1}{\sqrt{2j-1}} \right). \quad (9)$$

$k \geq 3$ かつ $m \geq 2$ のとき、(9) 式は (8) 式同様、次のように表わすことができる。

$$V_m \approx \sqrt{2cT} \left(\frac{k-m}{\sqrt{2m+1}} + \sqrt{2m-1/2} + \frac{1}{4\sqrt{2m-1}} - \sqrt{2k-5/2} - \frac{1}{\sqrt{2k-1}} - \frac{1}{4\sqrt{2k-3}} \right). \quad (10)$$

k が大きいとき、

$$V_m \approx \sqrt{2cT} \left(\frac{k-m}{\sqrt{2m+1}} + \sqrt{2m-1/2} - \frac{1}{4\sqrt{2m-1}} - \sqrt{2k} \right). \quad (11)$$

そこで、(7) ~ (9)、(11) 式から V_m/V を求め、図5に示した。図5から m 回以下の生起語について生起回数ごとに線形リストをまとめることにより、無駄領域が減少する様子が見える。特に、 m が小さいとき、減少の割合が高い。すなわち、低頻度キーワード領域に入れるのは、生起回数の小さい単語でよい。このことは、本稿で提案する方式が実用化に適していることを意味する。また、図5から k が大きくなると、減少度があまり変化しないことがわかる。そこで、 $k = \infty$ の場合の次式を簡単な評価式として利用することができる。

$$V_m/V \approx 1/\sqrt{2m+1}. \quad (12)$$

低頻度キーワードの線形リストにおけるリスト要素を固定長にし、このビット長を d とする。すると、 n 回

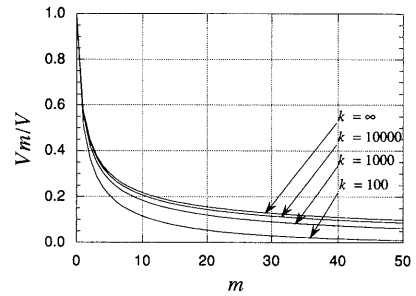


図5 無駄領域の減少度

Fig. 5 Decreasing degree of useless area.

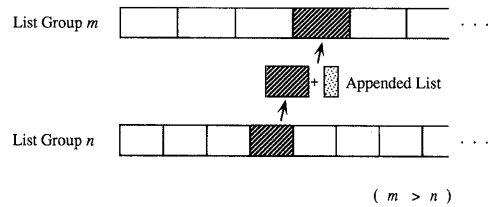


図6 生起回数の増加に伴う線形リストの移動

Fig. 6 Movement of linear list with increasing number of occurrences.

生起線形リスト群に必要な2次記憶領域量 L_n は、

$$L_n = dn I_n = \sqrt{2c} dn \left(\frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}} \right) \sqrt{T}. \quad (13)$$

また、全線形リスト群に必要な2次記憶領域量 $\sum_{n=1}^m L_n$ は、付録2の(vi)式より、次式のようになる。

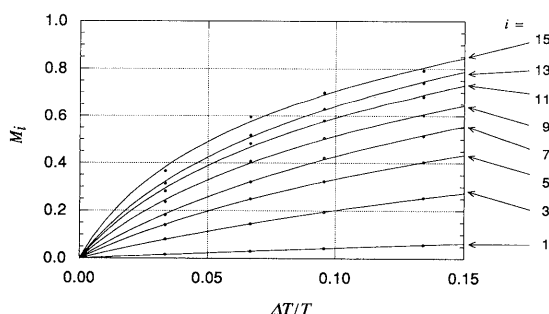
$$\sum_{n=1}^m L_n = \sqrt{2c} d \left(\sqrt{2m-1/2} + \frac{1}{4\sqrt{2m-1}} - \frac{m}{\sqrt{2m+1}} + a \right) \sqrt{T}. \quad (14)$$

高頻度キーワードのリスト領域の大きさが T に比例するのに対し、(14)式より低頻度キーワードのリスト領域の大きさは $\sqrt{T}$ に比例することがわかる。このことが、本稿の方式の一つの特徴である。

4.3 データ追加コスト

生起回数ごとにまとめて線形リストを管理すると、前節で述べたように、二次記憶領域の効率的利用という点で大きく改善される。しかし、この方式では文書の追加において、低頻度キーワードの生起回数が増加したとき、図6に示したように線形リストを別の領域に移動しなければならない。その移動コストは、移動するリストの数に比例すると考えられる。

延べ単語数 T の文書集合があるとき、 ΔT の延べ単語数をもつ文書集合が追加されたとき、 i 回生起単語が $j(j > i)$ 回生起となる確率 M_i についての数学モデルは知られていない。この M_i を i 回生起リスト領域

図7 移動度 M_i Fig. 7 Moving rate M_i .表3 a_i と b_i の値Table 3 Values of a_i and b_i .

i	a_i	b_i
1	0.231	2.140
2	0.475	3.101
3	0.405	6.468
4	0.549	6.429
5	0.429	11.703
6	0.503	11.054
7	0.518	12.835
8	0.488	15.484
9	0.444	21.882
10	0.449	24.843
11	0.441	28.195
12	0.401	37.015
13	0.461	30.179
14	0.420	39.937
15	0.404	47.117

から別の領域への移動の割合と考え、移動率(moving rate)ということにする。

表1, 表2のA ($T=87,354,577$) に対し, $\Delta T/T=0.0335, 0.0667, 0.0957, 0.1339$ としたときの移動率 M_i を図7に点で示した。図7からわかるように, 文書が13.4%増加したとき, 生起回数1の単語で生起が増加するものは6%に満たないのに対し, 生起回数15の単語では80%が増加する。

この移動率 M_i の近似式として次のようなものを考えた。

$$M_i \approx a_i \ln(b_i(\Delta T/T) + 1), \quad (15)$$

ここで, a_i と b_i は定数。

最小二乗法により求めた a_i, b_i を使って, 図7に(15)式を実線で示した。図7から, i が小さいとき(15)式の近似度は高いことがわかる。比較的小量の文書追加のような場合, $\Delta T/T$ が小さいため, M_i は(15)式より次式のように表わすことができる。

$$M_i \approx a_i b_i (\Delta T/T). \quad (16)$$

したがって, m 回以下生起する単語のリストをそれぞ

れまとめて管理した場合, リストの移動に伴い余分に必要となるコスト C_m^+ は

$$C_m^+ \propto \left(\sum_{i=1}^m a_i b_i I_i \right) \Delta T/T \quad (17)$$

となる。(17)式の $\sum_{i=1}^m a_i b_i I_i$ の計算の参考のために, 表3に図7に示した(15)式の a_i と b_i の値を示した。

5. 実現法

この章では, 本稿で提唱する方式の実現法について考察する。ここで想定している情報検索システムでは, INSPEC テープのように大量で定期的にデータ追加が行われている文書データを管理維持しているものとする。3章で述べた理由により, INSPEC テープ以外の英文科学技術2次文献データにも本稿の方式は適用可能であると考えられる。

本稿の方式では, m 回以下生起する低頻度キーワードの線形リストについては生起回数ごとに群として格納するわけであるが, この m は理論的に決定することができない。 m の増加に伴って, 領域量は減少し, 計算量は増加するが, このことから最適な m が求まるわけではない。なぜなら, 計算量の増加はデータ追加にかかわることであり, 検索性能とは無関係であるため, 領域量の減少と同列に比較する問題ではないからである。したがって, m を決める問題は, 情報検索システムの実現と管理運用の容易さと領域量の減少の程度との兼ね合いで決まることになる。

m を比較的小さくとした場合, リスト群に入らない低頻度キーワードの線形リストは4.1節で述べたような論理ブロックを用いた実現上の工夫が必要であるが, ここではこの問題に立ち入らない。

5.1 リスト群

まず, 図4に示した線形リスト群を実現するためには, データベース構築時に存在する文書集合の T と D から(3)式の $\sqrt{2C}$ を求める。次に追加量を推測し, 文書参照ファイルが扱う $T_{\max}$ を決定する。すると(3), (4)式から $T_{\max}$ に対する I_n が求まる。ここで, 4.2節のようにリストの要素を固定長にする。すると, (13)式に $T=T_{\max}$ とおくことにより, n 回生起線形リスト群に必要な2次記憶領域量が求まる。これから, 文書参照ファイルのブロック数が求まるので, このブロック数からなる連続領域を n 回生起リスト群のために割り当てる。

n 回生起キーワードの線形リスト群は, 長さ n の線形リストを q 個格納できる構造になる。 q は, 余裕をもたせて(3), (4)式と $T_{\max}$ で決まる I_n より大きく

とる。すなわち、

$$q = \sqrt{2c} \left(\frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}} \right) \sqrt{T_{\max}} + \beta,$$

ここで、 β は定数。

データ追加時には、このリスト群では、線形リストの消去と挿入が行われるので、 q 個の線形リスト格納領域が空いているかどうかの管理が必要である。この種の空き領域管理の技法は、すでに確立している¹⁾ので、ここではこの方式には触れない。

この方式では、 $T_{\max}$ を超えた場合、文書参照ファイルを再構築しなければならないが、このことはとりたてて欠点というわけではない。現実の情報検索システムでは、機能追加などにより、システムの再構築が必要になることが多いからである。また、検索効率の問題から文書の最大量を決めて運営している場合もあり、その場合、文書データを時期的に分割して別の文書データベースとして構築していることが多い。

T と D から求めた $\sqrt{2c}$ が $T_{\max}$ について適用することが妥当かという問題がある。INSPEC テープに関しては、 c は分野によって多少の異なりがみられるが、時間的には比較的安定しているようである。

5.2 索引

いま、低頻度キーワードに関して本稿の方式を採用しない場合を考える。このとき、線形リストは、物理ブロックあるいは論理ブロックのいずれかの先頭から置かれ、長い線形リストは複数のブロックを使って格納されるものであるとする。このとき、図1に示した索引の内容ポインターは、一般に二つのブロック番地をもつ必要がある。一つは、リストの先頭が置かれたブロック番地であり、検索時に必要である。もう一つは、リストの最終部分を格納したブロック番地であり、データ追加時にリストを延長するために必要である。ここでは、前者を先頭番地、後者を最終番地と呼ぶことにする。

本稿で提唱した方式を採用する場合、索引に関しては n 回生起低頻度キーワードの内容ポインタは、次のように変更すればよい。先頭番地には n 回生起リスト群の先頭ブロック番地を入れ、最終番地には n 回生起リスト群中における生起位置を入れる。リスト群にあるキーワードについては、その線形リストへアクセスするためのリスト先頭位置は、索引の内容ポインタの先頭番地から n が求まり、リスト要素が固定長であることとリスト群が連続領域に置かれていることから、最終番地によってこのリストの先頭位置を示すブロック番地とブロック内番地とを計算によって求めることができる。

データ追加時には、線形リスト位置が移動することがあるが、この場合の内容ポインタの変更法は自明であるので、述べる必要はないであろう。

6. む す び

大規模情報検索システムにおいて、高頻度キーワードと低頻度キーワードの線形リストを同じデータ構造で記憶する場合、高頻度キーワードリストの読み込みのために2次記憶アクセス回数を減らそうとすれば、巨大な無駄領域が生じることを示した。本稿では、 m 回以下生起する低頻度キーワードの線形リストについては、生起回数ごとにまとめて記憶する方法を提案し、この無駄領域を減少させることができることを示した。この量は、 m とともに減少する。一方、この方式では、文書追加時に生起回数が増加したキーワードのリストを別の領域に移動するために余分なコストがかかる。このコストは m とともに増加する。

本稿の解析は、 m が比較的小さい場合でも無駄領域が大きく減少することを示している。例えば、 $m=4$ でも無駄領域の2/3を除去できる。このことは、実用のためには都合のよい特性である。

高頻度キーワードの線形リストの長さは文書量に比例するのに対し、本稿の方式によって低頻度キーワードのリストを管理すれば、必要な領域は文書量の平方根に比例する量ですむ。

参 考 文 献

- 1) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass. (1983).
- 2) Bayer, R. and McCreight, E.: Organization and Maintenance of Large Ordered Indexes, *Acta Inf.*, Vol. 1, No. 3, pp. 173-189 (1972).
- 3) Comer, D.: The Ubiquitous B-tree, *Comput. Surv.*, Vol. 11, No. 2, pp. 121-137 (1979).
- 4) Salton, G. and McGill, M. J.: *Introduction to Modern Information Retrieval*, McGraw-Hill, New York (1983).
- 5) Zipf, G. K.: *Human Behaviour and the Principle of Least Effort*, Addison-Wesley, Cambridge, Mass. (1949).
- 6) Shannon, C. E.: Prediction and Entropy of Printed English, *Bell Syst. Tech. J.*, Vol. 30, pp. 50-64 (1951).
- 7) Aithison, T. M., Martin, M. D. and Smith, J. R.: Developments towards a Computer Based Information Services in Physics, Electrotechnology and Control, *Inform. Storage and Retrieval*, Vol. 4, No. 2, pp. 177-186 (1968).

- 8) Matsuo, F.: 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).
- 9) Booth, A. D.: A "Law" of Occurrences for Words of Low Frequency, *Inform. Control*, Vol. 10, No. 4, pp. 386-393 (1967).
- 10) 二村祥一, 松尾文碩: 英文科学技術文献情報に対する不要語除去法による自動索引, 情報処理学会論文誌, Vol. 28, No. 7, pp. 737-747 (1987).

付 録 1

順位 r の生起頻度を $f(r)$ とすると, n 回生起する単語の順位 r については, 次式が成立すると考えることができる.

$$n-1/2 \leq f(r) < n+1/2. \quad (\text{i})$$

(i) 式において $f(r) = Tp(r) = cT/r^2$ とおくと,

$$\sqrt{cT/(n+1/2)} < r \leq \sqrt{cT/(n-1/2)}. \quad (\text{ii})$$

ゆえに,

$$\begin{aligned} I_n &= \sqrt{cT/(n-1/2)} - \sqrt{cT/(n+1/2)} \\ &= \sqrt{2cT} \left(\frac{1}{\sqrt{2n-1}} - \frac{1}{\sqrt{2n+1}} \right). \end{aligned} \quad (\text{iii})$$

また, (ii) 式において $n=1$ とおくと

$$r \leq \sqrt{2cT}. \quad (\text{iv})$$

$D=r_{\max}$ と考えることができるので, (3) 式がえられる. また, (iii) 式に (3) 式を代入し, (4) 式をえる. これは, Zipf が I_n と D を導いた方法であるが, Zipf は $p(x)$ に (1) 式を用いたため, えられた式は近似度が悪かった⁹⁾.

付 録 2

(8) 式と (10) 式を導くためには,

$$S(n) = \sum_{i=1}^n \frac{1}{\sqrt{2i-1}}$$

の近似式が必要である. いま,

$$U(n) = \sum_{i=1}^n \frac{1}{\sqrt{2i}}$$

とすると,

$$\begin{aligned} S(n+1) + U(n) &= \sum_{i=1}^{2n+1} \frac{1}{\sqrt{i}} \\ &\approx \int_{1/2}^{2n+3/2} \frac{dx}{\sqrt{x}} \\ &= 2(\sqrt{2n+3/2} - 1/\sqrt{2}). \end{aligned} \quad (\text{i})$$

ところで,

$$U(n+1) + 1 < S(n), \quad (\text{ii})$$

$$1/\sqrt{2n+1} + U(n) < S(n+1). \quad (\text{iii})$$

(ii) 式と (iii) 式から

$$1/\sqrt{2n+1} < S(n+1) - U(n) < 1. \quad (\text{iv})$$

(i) 式と (iv) 式から

$$\begin{aligned} \sqrt{2n+3/2} - 1/\sqrt{2} + 1/\sqrt{2n+1} &< S(n+1) \\ &< \sqrt{2n+3/2} - 1/\sqrt{2} + 1/2. \end{aligned} \quad (\text{v})$$

いま,

$$B(n) = \sqrt{2n+3/2} - 1/\sqrt{2}$$

とおくと, $S(n+1)$ は $B(n) + 1/2\sqrt{2n+1}$ と $B(n) + 1/2$ の間にある. この中間値は, $B(n) + 1/4\sqrt{2n+1} + 1/4$ であるが, $S(n+1)$ はこの値より少し大きい. そこで, 近似式として,

$$S(n+1) \approx B(n) + 1/4\sqrt{2n+1} + 1/4 + 0.0207$$

とした. これを書き直すと次式をえる.

$$S(n) \approx \sqrt{2n-1/2} + 1/4\sqrt{2n-1} - \alpha, \quad (\text{vi})$$

ここで, $\alpha = 1/\sqrt{2} - 1/4 - 0.0207 = 0.43640678\cdots$.

(vi) 式の評価を表 4 に示す. 表 4 から, $n \geq 2$ では (vi) 式の近似度が高いことがわかる.

(平成 6 年 11 月 17 日受付)

(平成 7 年 3 月 13 日採録)

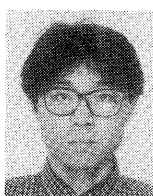
表 4 $S(n)$ の近似値
Table 4 Approximation of $S(n)$.

n	real value	approximate value	absolute error	relative error
1	1.00000000	1.04693809	0.046938	0.046938
2	1.57735027	1.58735948	0.010009	0.006346
3	2.02456386	2.02920450	0.004641	0.002292
4	2.40252834	2.40529713	0.002769	0.001152
5	2.73586167	2.73773355	0.001872	0.000684
6	3.03737302	3.03873605	0.001363	0.000449
7	3.31472311	3.31576536	0.001042	0.000314
8	3.57292200	3.57374688	0.000825	0.000231
9	3.81545763	3.81612726	0.000670	0.000176
10	4.04487336	4.04542759	0.000554	0.000137
50	9.57231373	9.57228784	0.000026	0.000003
100	13.71442242	13.71436214	0.000060	0.000004
500	31.19504999	31.19497279	0.000077	0.000002

**松尾 文碩 (正会員)**

昭和 16 年生. 昭和 41 年九州大学大学院工学研究科電子工学専攻修士課程修了. 工学博士. 九州大学工学部電気工学科勤務. 推論機構, 自然言語理解, データベース, 情報検索

システム, エキスパートシステムの研究に従事.

**高山 悟 (正会員)**

昭和 44 年生. 平成 6 年九州大学工学部電気工学科卒業. 九州大学大学院工学研究科電気工学専攻在籍中. 研究テーマは情報検索システム.

**佐藤 誉夫 (正会員)**

昭和 46 年生. 平成 7 年九州大学大学院工学研究科電気工学専攻修士課程修了. 日立超 LSI エンジニアリング (株) 勤務. 在学中の研究テーマは情報検索システム.
