データベースシステム 119−16
（1999. 7. 21）

ベクトル空間圧縮モデルによるWWW検索処理の効率化

原田晃史[†1] 川越恭二[†2]

インターネットの普及により急激に増大したWWW集合から効率的に必要な情報を取り出す必要性が増している．そこで，膨大に生成・蓄積されたWWW集合から，必要なWWW情報を効率的に取り出すシステムが必要となってきている．この要求を満たすために，ベクトル空間モデルに次元数を減少するための改良を加えたモデルを提案する．このモデルを用いて関連性のあるWWW集合をクラスタとしてまとめ，更にその類似度によってページ割り当てを決定する格納構造により，検索処理の効率化を図る．本方法では，格納構造を階層化し，この木構造を検索時に辿ることで，必要な情報を格納しているページを容易に決定することができる．また，格納時のオーバーフロー処理についても動的に対応できる方法である．シミュレーションの結果，本方法は，問い合わせに含まれるキーワード数が多い場合の検索や，ベクトル全体の関連性の強度が高い情報集合に対する検索に適していることが得られた．

The efficient WWW retrieval process based on Vector Space Compression Model

KOJI HARADA[†1] and KYOJI KAWAGOE[†2]

The paper presents new index structure that modifies Vector Space Model, in order to reduce the size of vector space dimension and to improve the number of page access under compression. We intend to develop this model to realize an efficient WWW directory service as a physical structure in searching appropriate page given a set of keywords. The paper describes the basic structure and some algorithms for insertion and searching. In the model, each vector in the vector space is represented only by a list of the keyword number where the document contains the keyword, not by the whole vector elements. The model is also evaluated in the three viewpoints: space efficiency, the number of page accesses for searching and the number of page accesses for insertion. It is shown from some evaluation that in the model the number of page accesses for searching is almost constant as the number of keywords given varies.

1. はじめに

インターネットの普及により急激に増大したWWW（World Wide Web）集合から効率的に必要なWWW情報を取り出す必要性が増大している．そこで，膨大に生成・蓄積されたWWW集合から，必要な情報を効率的に取り出すシステムが必要となってきている．このため，索引ファイルや関係データベースを用いた検索エンジンも実用化されているが，より効率的な手法のアプローチの1つとして，ベクトル空間モデル[1),2)]による手法がある．2次記憶装置にWWW検索のための索引情報を格納しているページを検索時に参照することをページアクセスと呼ぶことにすると，索引ファイルでは，1つのキーワードに対して少なくとも1回のページアクセスが必要であり，質問として複数のキーワードが与えられると，検索に要するページアクセス回数は増大する．これに対して，ベクトル空間モデルは，質問やドキュメントをベクトルで表現するため，複数キーワードが与えられたときの検索に適していると考えられる．しかし，ベクトル空間モデルの問題点の1つに，ベクトルの次元が大きくなるという問題がある．これまで，ベクトルの次元を減少させるために，特異値分解[3)]を適用する方法が提案されているが，数十から数百までしか減少しない．

本論文では，ベクトル空間モデルを改良してベクトルの次元数を圧縮する方法(ベクトル空間圧縮モデル：以降，提案拡張モデル）を提案する．さらに，提案拡張モデルを用いて検索効率の向上を図るために，2次記憶装置内での格納構造とその内部処理を提案する．

†1 立命館大学大学院理工学研究科
Graduate School of Science and Engineering, Ritsumeikan Univ.
†2 立命館大学理工学部
Faculty of Science and Engineering, Ritsumeikan Univ.

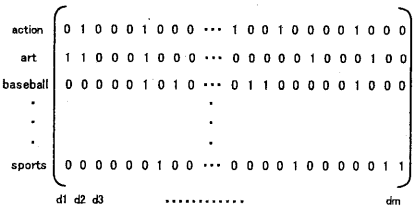

図1 ベクトル空間モデルの行列表現例
Fig.1 Example of Vector Space Model with matrix presentation

2. ベクトル空間モデル

2.1 概要

ベクトル空間モデルでは，質問とドキュメントの両方を，キーワードを軸とした多次元空間におけるベクトルとみなす．予め，ドキュメント集合中のキーワードをすべて列挙し，その各々について各ドキュメントとの適合度（ドキュメントにそのキーワードが含まれないなら 0，そうでなければ 1 或いは正の実数値）を算出し，そのドキュメントのベクトルとする．

ベクトル空間モデルの行列表現の例を図 1 に示す．個々のベクトルは，次のように表すことができる．

$$\vec{d}_i = {}^T[w_1^i, w_2^i, w_3^i, \cdots, w_N^i]$$

なお，厳密には，次の式で表すことができる．

$$\vec{d}_i = [I, J, K, \cdots, Z]^T [w_1^i, w_2^i, w_3^i, \cdots, w_N^i]$$
$$= w_1^i I + w_2^i J + w_3^i K + \cdots + w_N^i Z$$

ここで，$I, J, K, \cdots, Z$ （N 個）は正規直交系をなす単位ベクトル，$w_n^i, n = 1,2,\cdots N$ はドキュメントベクトル $\vec{d}_i$ とキーワードとの適合度，T は転置行列を示す．

2.2 問題点とその対策

ベクトル空間モデルの問題点は，個々のベクトルの次元が N 次元と高くなることである．ここで N とは，全てのキーワードの総数である．このモデルを図 1 のような行列で表現すると一般に疎行列となる．ベクトル空間モデルを WWW 集合の検索に適用することを考えると，個々の WWW ページ（ドキュメント）のキーワードとして設定されるものは，数個～10 数個程度であることを考えると，ドキュメントに関連のあるキーワードだけをベクトルの要素とすることにより，ベクトルの次元を減少させることができる．本論文では，このようにベクトル空間モデルを改良する．すなわち，単位ベクトル $I, J, K, \cdots, Z$ をキーワードの識別番号（軸番号）と考え，ドキュメントのキーワードとなる番号のみを $h^i(p)$ に格納する．これにより，個々のベクトルの次元数を大幅に減少させることができる．

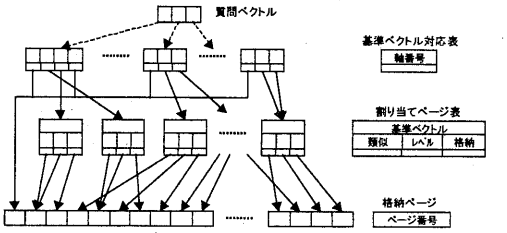

図2 基本的格納構造
Fig.2 Basic physical structure

ここで，n_i：ドキュメントのキーワード数，w_p^i：ドキュメントとキーワードの適合度とすると，各ドキュメントのベクトルは次のように表現できる．

$$\vec{d}_i = \sum_{p=1}^{n_i} w_p^i h^i(p) \qquad (1)$$

このように定義すると，個々のベクトルの次元数は，そのベクトルの持つキーワードの個数となるので，従来の方法よりもベクトルの次元をかなり減少できる．

3. 提案拡張モデルによる効率的な検索方法

3.1 主な考え方

検索の効率を向上させるために，2 次記憶装置内に格納すべきページを次のように定義する．すなわち，以降で定義する類似関数を用いたベクトルの類似度によってまずクラスタを生成し，更にその類似度によってクラスタ内を分類し，ページを割り当てる．ここで，クラスタの代表元のことを基準ベクトルと呼ぶことにする．

このように分類することによって，検索時は，質問ベクトルと基準ベクトルとの比較のみによって目的のページに辿り着ける．しかし，このままでは検索時に質問ベクトルと基準ベクトルとの総当りになり，検索処理時間の増加をまねく．そこで，基準ベクトルと，それとの類似度による割り当てページへのポインタとを保持する表（以下，割り当てページ表：基準ベクトルとの類似レベル毎のページ位置を示す）と，キーワード毎に対応する基準ベクトルを格納している表へのポインタとを保持する表（以下，基準ベクトル対応表：クラスタの存在位置を示す）を作成し，検索効率の向上を図る．

図 2 に，本方法の基本的格納構造を示す．ここでは，類似度による区分数を 3，ドキュメントのキーワード数を 3 としている．

3.2 諸定義

以下に，本論文で用いる概念を定義する．

3.2.1 ベクトルの内積

ベクトルの表現を(1)式のように表現すると，ベクトル間の内積は次のように表すことができる．

$$\vec{d}_i \circ \vec{d}_j = \sum_{p=1}^{n_i} \sum_{q=1}^{n_j} w_p^i h^i(p) \cdot w_q^j h^j(q) \qquad (2)$$

但し $\begin{cases} h^i(p) \cdot h^j(q) = 1 & if\ h^i(p) = h^j(q) \\ h^i(p) \cdot h^j(q) = 0 & if\ h^i(p) \neq h^j(q) \end{cases}$

3.2.2 ベクトルの類似関数

類似関数として，コサイン関数を用いる．ベクトルのノルムをその要素数の平方根と考え，そしてベクトル間の内積を(2)式で定義すると，ベクトル間の類似関数 SIM は次のように表すことができる．

$$SIM(\vec{d}_i, \vec{d}_j) = (\vec{d}_i \circ \vec{d}_j) / (\|\vec{d}_i\| \cdot \|\vec{d}_j\|)$$
$$= (\vec{d}_i \circ \vec{d}_j) / (\sqrt{n_i} \cdot \sqrt{n_j}) \qquad (3)$$

3.2.3 ベクトルの距離関数

ベクトル間の距離を，その 2 ベクトルを含む平面における単位円と 2 ベクトルとの交点のユークリッド距離とする．3.2.2 より類似関数としてコサイン関数を用いているので，ベクトル間の距離関数は次のように表すことができる．

$$D(\vec{d}_i, \vec{d}_j) = \sqrt{2 \cdot (1 - SIM(\vec{d}_i, \vec{d}_j))} \qquad (4)$$

3.2.4 基準ベクトル

クラスタの代表元のことを基準ベクトルとする．あふれページ内の任意のベクトルにおいて，その他のベクトルとの距離の総和が最小となるものが，クラスタの重心に最も近いベクトルとなるため，この条件を満たすベクトルを基準ベクトルとして選択する．ベクトル間の距離関数を(4)式で定義すると，基準ベクトルを導出する式は次のように表すことができる．

$$\vec{e}_k = \vec{d}_i \ such\ that\ \min_i \left(\sum_{j=1}^{m} D(\vec{d}_i, \vec{d}_j) \right) \qquad (5)$$

3.2.5 拡張基準ベクトル

クラスタ内のベクトルの要素集合から 1 つ選び，それを基準ベクトルの要素として加えたものを拡張基準ベクトルと呼ぶ．すなわち，オーバーフローページのベクトルの要素集合を N^f，$\exists w_p^i h^i(p) \in N^f$，$h^k(q) \neq h^i(p)$ $for\ \forall h^k(q), q = 1 \cdots n_k$，かつ出現回数の少ないものとすると，拡張基準ベクトルは次のように表すことができる．

$$\tilde{\vec{e}}_k = \vec{e}_k + w_p^i h^i(p) = \sum_{q=1}^{\tilde{n}_k} w_q^k h^k(q) \qquad (6)$$
$$(\because w_{n_k+1}^k h^k(n_k+1) = w_p^i h^i(p), \tilde{n}_k = n_k + 1)$$

3.3 格納方法

初めに，格納すべきベクトルの各要素を用いて基準ベクトル対応表を調べ，基準ベクトルを確定する．ここで，基準ベクトルが確定しない場合は，格納対象ページをあふれページとし，あふれページに格納する．

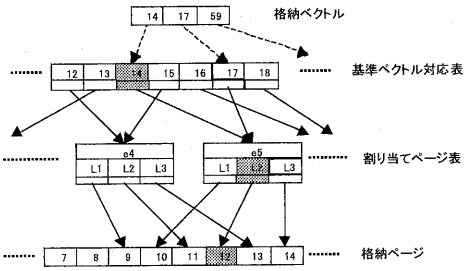

図 3 格納方法の例
Fig.3 Example of insertion process

次に，格納ベクトルと基準ベクトルとの類似度を算出し，その値を用いて割り当てページ表を調べ，格納対象ページを確定する．

このとき，格納対象ページに格納されているベクトル数が 1 ページ内の最大ベクトル数（以降，ベクトルの許容数と呼ぶ）未満である場合は，そのページにベクトルを格納する．

図 3 に，格納方法の例を示す．ここで，類似度による区分数を 3（その類似区分を L1，L2，L3 とする．但し，0<L1<=0.4，0.4<L2<=0.7，0.7<L3<=1 とする．），ドキュメントのキーワード数 n を 3，ドキュメントとキーワードの適合度 w_p^i を 1 とする．

例えば，登録する WWW ページのキーワードとして，立命館大学（軸番号：14），理工学部（軸番号：17），入学案内（軸番号：59）と入力すると，次のように格納される．

[1] 格納ベクトル $\vec{d}$ =(14,17,59) を作成．
[2] 格納ベクトルの各要素について基準ベクトル対応表を調べ，基準ベクトル $\vec{e}_s$ を確定．
[3] (3)式を用いて格納ベクトルと基準ベクトルとの類似度 $SIM(\vec{d}, \vec{e}_s)$ を算出し，類似区分 L2 を決定．
[4] 割り当てページ表からその類似区分のページ（p12）を確定し，ドキュメントを格納．

格納対象ページに格納されているベクトル数がベクトルの許容数と一致している場合，すなわちオーバーフローが発生する場合には，次の 3 通りが考えられる．

(A) 格納対象ページがあふれページのとき
(B) 格納対象ページが基準ベクトルとの類似度が最も高いページ（すなわち，クラスタ内の最上位ページ）のとき
(C) 格納対象ページが上記以外のページ（すなわち，クラスタ内の最上位ではないページ）のとき

ここで，(A)の場合にはクラスタリング，(B)の場合にはクラスタグリッド，(C)の場合にはシャッフリングというページ分割アルゴリズムを提案し，このアルゴ

リズムを用いてベクトルを格納しなおす．

また，複数のページで同時にオーバーフローが発生する場合も考えられるが，そのときには，オーバーフローが発生したクラスタ内の上位ページから順にそのページに適したページ分割アルゴリズムを適用する．

3.4 ページ分割アルゴリズム

3.4.1 クラスタリング

格納ベクトルを含むあふれページ内の任意のベクトルで，他のベクトルとの距離の総和が最小となるものを基準ベクトルとして選び，この基準ベクトルと類似性を持つベクトルをクラスタとしてまとめ，新しいページに割り当てる．すなわち，(5)式から基準ベクトルを求め，$\forall \vec{d}_f \in \Omega$ において $SIM(\vec{e}_k,\vec{d}_f) \succ 0$ を満たすものを新しいページに格納する．ここで，Ω：オーバーフローが発生したページ内の任意のベクトル集合とする．

3.4.2 クラスタグリッド

クラスタの基準ベクトルとの類似度によって，クラスタ内のページを 2 分割する．分割の指標となる類似度は，小さいものから順に上げていく．すなわち，$\forall \vec{d}_f \in \Omega$ において $SIM(\vec{e}_k,\vec{d}_f) \succ \delta$ を満たすものを新しいページに格納する．ここで，Ω：オーバーフローが発生したページ内の任意のベクトル集合，δ：指標となる類似度とする．

3.4.3 シャッフリング

格納ベクトルとオーバーフローが発生したページ内のベクトルの要素から 1 つを選び，これをクラスタの基準ベクトルの要素として追加し，クラスタ内の任意のベクトルを再分類する．すなわち，(6)式から拡張基準ベクトルを求め，$\forall \vec{d}_f \in \Phi$ において $SIM(\vec{e}_k,\vec{d}_f)$ を算出して，ページを割り当てしなおす．ここで，Φ:オーバーフローが発生したページを含むクラスタの任意のベクトル集合とする．

3.5 検索方法

まず，質問ベクトルの各要素を用いて基準ベクトル対応表を調べ，基準ベクトルを確定する．このとき，複数の基準ベクトルを算出可能である．ここで，基準ベクトルが確定しない場合は，検索対象ページをあふれページ（どのクラスタにも属さないドキュメントを格納するページ）とする．

次に，質問ベクトルと基準ベクトルとの類似度を算出し，その値を用いて割り当てページ表を調べ，その類似度以上のページを検索対象ページとし，そのページ内の任意のベクトルについて，質問ベクトルを完全に満たすものを出力する．

図 4 に，検索方法の例を示す．ここで，類似度による区分数を 3 とし，その類似区分をそれぞれ L1，L2，

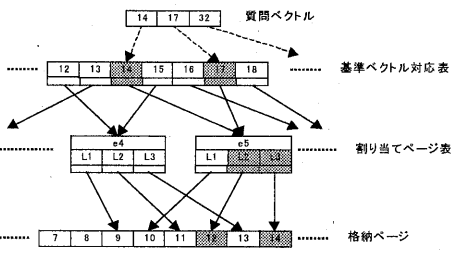

図 4 検索方法の例
Fig.4 Example of search process

L3 とする．但し，0<L1<=0.4, 0.4<L2<=0.7, 0.7<L3<=1 とする．また，ドキュメントのキーワード数 n を 3，ドキュメントとキーワードの適合度 w_p^i を 1 とする．

例えば，質問に用いるキーワードとして，立命館大学（軸番号：14），理工学部（軸番号：17），情報学科（軸番号：32）と入力すると，次のような検索が実行される．ここで，n は質問やドキュメントが保持できる最大キーワード数としている．

[1] 質問ベクトル $\vec{q}$=(14,17,32) を作成．

[2] 質問ベクトルの各要素について基準ベクトル対応表を調べ，基準ベクトル $\vec{e}_s$ を確定．

[3] 質問ベクトルと基準ベクトルとの類似度 $SIM(\vec{q},\vec{e}_s)$ を算出し，類似区分 L2 を決定．

[4] 割り当てページ表からその類似区分以上のページ（p12，p14）を確定．

[5] p12，p14 に格納されているすべてのベクトル $\vec{d}_i$ において，$SIM(\vec{q},\vec{d}_i)$>=3/n を満たすものを出力．

4. 評価実験

本方法を評価するために，ドキュメントの関連強度を変化させて評価する実験と登録ドキュメント数を変化させて評価する実験を行う．これらシミュレーション実験の評価項目には，平均ページアクセス回数と空間使用効率[4]の 2 つを使用し，それぞれ次のように定義する．

- 平均ページアクセス回数
  Σ（ページアクセス回数）/シミュレーション回数
  これは，最も時間のかかる 2 次記憶装置とのファイルアクセスを推定する値であり，検索処理時間に反映されるので，検索処理の効率を図る評価値として用いる．
- 空間使用効率
  （データサイズ*データ件数）/（ページサイズ*ページ数）

=（データ件数/ページ数）*（1/M）

（但し，Mは1ページ内のベクトル許容数を示す）

これは，2次記憶装置内の物理格納空間のどのくらいを使用しているかという割合であり，100%が理想的である．索引ファイルとして通常用いられるB-木は，平均して約69%の空間使用効率となっている[4),5)]．

また，ドキュメントを1つ登録する毎のページの更新・作成回数を評価する実験も行う．

4.1 実験1：ドキュメント間の偏りが性能に与える影響

ドキュメント全体の関連強度を変化させ，平均ページアクセス回数と空間使用効率を評価する．なお，ドキュメントの関連強度とは，ドキュメント間の偏りの程度のことを意味する．

実験に用いたドキュメント数は100，キーワード数は270とし，この中から6つをランダムにベクトルに割り当てた．この割り当て方法を使用して以下の3つのcaseを与える．case1からcase3への順にベクトル全体の関連性の強度を順に高めている．

- case1：ベクトル集合を大きく2つに分ける．
- case2：ベクトル集合を大きく3つに分ける．
- case3：ベクトル集合を大きく4つに分ける．

また，ドキュメントとキーワードの適合度を1，ページのベクトル許容数Mを5とする．

上記の条件の下でシミュレーション実験を行い，各caseでの空間使用効率と，通常WWW検索時に利用者が質問入力するキーワード数を1～6と想定して変化させたときの平均ページアクセス回数を算出する．

シミュレーションを行った結果を図5に示す．空間使用効率に関しては，case1では74.1%，case2では76.9%，case3では76.9%となり，平均すると約76%となった．なお，図5では各caseにおいて10回検索したときの平均値を示した．

4.2 実験2：ドキュメント数とキーワード数の変化が性能に与える影響

登録ドキュメント数を変化させ，ページのベクトル許容数を固定した場合の平均ページアクセス回数を評価する．

実験に用いたドキュメント数は，100～1000で100個単位で増加させた．各々のキーワード数は，160～2100で，登録ドキュメント数が500のときまでは約150個，それ以上のときは約250個単位で増加させ，この中から6つをランダムにベクトルに割り当てた．また，ドキュメントとキーワードの適合度を1，質問に含まれるキーワード数を4，1ページ内のベクトル許容数Mを10と固定した．このような条件の基で，検索時における平均ページアクセス回数を算出する．

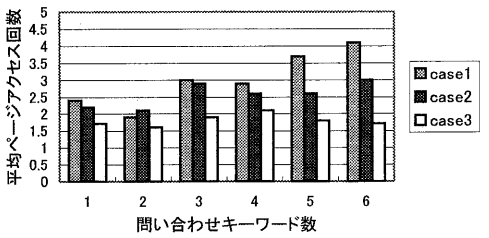

図5 平均ページアクセス回数

Fig.5 Estimation of the average of the number of page accesses for searching on all cases

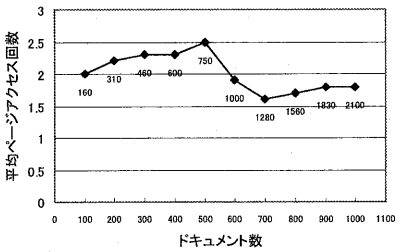

図6 ドキュメント数に対する平均ページアクセス回数

Fig.6 The average of the number of page accesses for searching for the number of documents

シミュレーションを行ったときの登録ドキュメント数毎の10回検索を行ったときの平均ページアクセス回数の結果を図6に示す．なお，図中のデータラベルの値は，使用したキーワードの総数を示す．

本方法では，登録ドキュメント数が増加しても，1.5～2.5回のページアクセスで目的のドキュメントを検索することができた．また，キーワード数を約250単位で増加させたときは，約150単位で増加させたときに比べて，若干平均ページアクセス回数が減少した．

4.3 実験3：ドキュメント数とページのベクトル許容数の変化が性能に与える影響

登録ドキュメント数とページのベクトル許容数を変化させ，平均ページアクセス回数と空間使用効率を評価する．

実験に用いるドキュメント数を100，500，1000とする．各々のキーワード数数は160，750，2100とし，この中から6つをランダムに個々のドキュメントのベクトルとして割り当てた．また，ドキュメントとキーワードの適合度を1，質問に含まれるキーワード数を4と固定して，1ページ内のベクトル許容数Mを5，7，10と変化させる．このような条件を使用して，空間使用効率と平均ページアクセス回数を算出する．

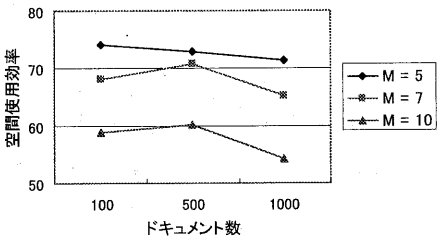

図7 ドキュメント数に対する空間使用効率の変化
Fig.7 Space efficiency for the number of documents

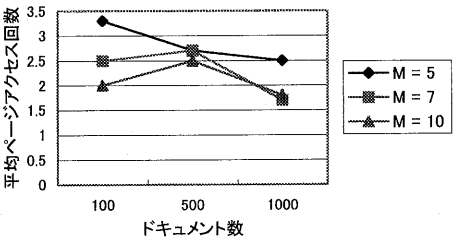

図8 ドキュメントに対する平均ページアクセス回数の変化
Fig.8 The average of the number of page accesses for searching for the number of documents

シミュレーションを行ったときの登録ドキュメント数毎の空間使用効率の変化を図7に，各条件の基で10回検索したときの平均ページアクセス回数の変化を図8に示す．

4.4 実験4：ドキュメント登録時の性能

ドキュメントを1つ登録する毎のページの更新・作成回数を評価する．

実験に用いるドキュメント数を1000とする．キーワードの総数を2000とし，この中から6つをランダムに個々のドキュメントのベクトルとして割り当てた．また，ドキュメントとキーワードの適合度を1，1ページ内のベクトル許容数Mを10と固定する．このような条件で，ドキュメントを1件ずつ登録し，そのときに更新・作成されるページ個数の最小値，最大値，平均値を算出する．

シミュレーションを行った結果，作成・更新回数に対するデータ件数の分布を図9に示す．図9に示すページの更新・作成回数は，最小で1回，最大でも4回となり，平均すると約1.3回であった．また，作成・更新回数が3回であったものは3件，最大の4回であったものは4件であり，全体の1%未満であった．

5. おわりに

本論文では，ベクトル空間モデルに改良を加えた拡張モデルを提案し，この拡張モデルを用いて検索処理

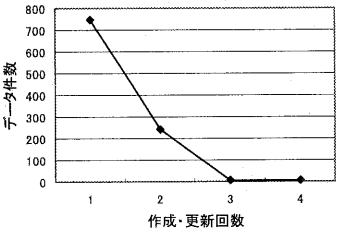

図9 ページの作成・更新回数に対するデータ件数の分布
Fig.9 Distribution of document data for necessary page accesses for insertion

を効率的に行う方法を説明した．シミュレーションを行った結果，本方法は，質問に含まれるキーワード数が多い場合の検索や，ベクトル全体の関連性の強度が高い情報集合に対する検索に適していることが得られた．WWW集合は，ある程度まとまりのある偏った集合であると考えられるため，本方法はWWW検索処理を効率的に行うことができるものと考えられる．

最後に，今後の課題を以下に列挙する．

- ドキュメントとキーワードとの適合度に重み付けを行った実験による評価
- 実際のWWW検索への実装
- 従来の索引ファイルとの性能比較

参 考 文 献

1) Christos Faloutsos: SEARCHING MULTIMEDIA DATABASES BY CONTENT, Kluwer Academic Publishers, 1998.
2) A.Tomasic, L.Gravano, C.Lue, P.Schwarz and L.Haas: Data Structures for efficient Broker Implementation, ACM Trans. on Information Systems, Vol.15, No.3, PP.223-253, July 1997.
3) 渡辺 正裕, 石川 佳治, 吉川 正俊, 植村俊亮: 多次元ベクトルの視覚的探索機能を有する情報検索, 情報処理学会データベースシステム研究会研究報告, 第96-DBS-109巻, pp.7-12, July 1996.
4) 北川 博之: データベースシステム, 情報系教科書シリーズ14巻, 昭晃堂, 1996.
5) Gio Wiederhold: File Organization for Database Design, McGraw-Hill, 1987.
6) 原田 晃史, 川越 恭二: ベクトル空間圧縮モデルによるWWW検索処理の効率化, 情報処理学会第58回全国大会講演論文集, vol.3, 3T-06(1999).