データベースシステム 113－32
（1997. 7. 15）

高速テキスト検索のための パトリシアトライ構造化シグネチャファイル

権藤 夏男，金子 邦彦，牧之内 顕文
九州大学大学院システム情報科学研究科
〒812–81 福岡市東区箱崎 6-10-1
gondo@db.is.kyushu-u.ac.jp,{kaneko,akifumi}@is.kyushu-u.ac.jp

指定されたキーワードを含むようなテキストの検索（全文検索）のためのインデックスは，現在でも重要な研究テーマである．全文検索用インデックスの一種であるシグネチャファイルのサイズ及び全文検索処理コストは，後処理であるフォルスドロップ (false drop) の処理コストを無視すると，検索対象のテキスト数に比例する．シグネチャファイルをビットデータ用探索木の一種であるパトリシアトライ (Patricia Trie) と組み合わせると，我々の実験では，そのサイズは本来のシグネチャファイルとおおよそ同じであり，その全文検索処理コストは，検索対象のテキスト数でなく，検索結果として得られるテキスト数と相関があることが分かった．その結果，多数のテキストを絞り込んで少数のテキストを得るような場合に有効であるとの結論を得た．
キーワード： 全文検索，インデックス，シグネチャファイル，トライ，テキストデータベース

Patricia Trie Structured Signature File for Full Text Search

Natsuo GONDO, Kunihiko KANEKO, and Akifumi MAKINOUCHI
Graduate School of Information Science and Electrical Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-Ku Fukuoka-Shi, 812-81,
gondo@db.is.kyushu-u.ac.jp, {kaneko,akifumi}@is.kyushu-u.ac.jp

Full-text search is important research issue. Full-text search is to search the documents that contain the specified keyword(s). Signature file is one of the effective indexes for full-text search. The size of signature file and the retrieval cost using signature file increase in proportion to the number of documents in a database. In this paper, we propose a Patricia trie structured signature file. From our experiment, the number of retrieved documents and the retrieval cost using the Patricia trie structured signature file correlates. We conclude that our index is practical when the size of database is large, and the number of retrieved documents is relatively small.
Keywords: full-text search, index, signature file, trie, text database

1 はじめに

テキストデータの検索は重要な研究課題である．これは，インターネット上の World Wide Web (WWW) の普及などにより，多量のテキストデータが計算機上に蓄積されるようになったことが背景にある．

テキストデータの検索法としては，全文検索が最も重要である．全文検索は，指定されたキーワード（検索文字列）を含むようなテキストを探し出すというものである．テキストデータの検索法は，（1）利用者にテキストデータごとの意味付け（キーワード等）を強いるものと，（2）そうでないものとの２種類に大別することができるが，後者の方が利用者に負担をかけないという点で優れている．全文検索も後者に属する．

全文検索を高速に処理するには，インデックスが有効である．従来，全文検索を高速に行うためのインデックスとして，シグネチャファイル (signature file)，転置ファイル (inverted file)，シグネチャファイルの改良としての文字成分表などの研究が行われてきた．シグネチャファイルと転置ファイルのサイズを比較すると，シグネチャファイルの方が小さく，優れている．

シグネチャファイルの課題の１つは，全文検索時におけるシグネチャファイル走査コストの削減である．シグネチャファイルによる全文検索の原理は，あらかじめテキストごとにテキストシグネチャと呼ばれる固定長（一般には１０２４ビット）のビット列をある規則により作成しておくと，テキストシグネチャのみを走査することで検索文字列を含むようなテキストの候補を精度よく選び出すことができることにある．問題は，全文検索において全テキストシグネチャの走査が必要なことであり，多数のテキストを検索対象とするとき，走査コストが増大する．

全文検索時におけるシグネチャファイル走査コストの削減のために，いくつかのシグネチャファイルの分割格納法が提案されてきた [5]．その代表的な手法は，次の２つにまとめられる．（1）各テキストシグネチャを数ビットずつの部分シグネチャに分割し，全テキストの部分シグネチャをまとめて格納する方法と，（2）先頭数ビットなどの値をもとに，シグネチャファイルを分割格納する方法である．いずれにしても，従来の研究の多くでは，シグネチャファイルのサイズは実メモリよりも大きい，すなわちシグネチャファイルの一部または大部分がディスク上に置かれていることを仮定しており，分割格納により走査時におけるディスクＩ／Ｏコストを削減することを主眼としていた．これは，ディスクＩ／Ｏをボトルネックととらえ．シグネチャファイルをいくつかの集まりに分割してディスクページを意識して格納することで，結果として読み出されるディスクページ数を削減するものである．

近年，メモリの低価格化，高速な全文検索への要求により，シグネチャファイルをすべてメモリ上に置くことが現実的になってきた．そこで本論文では，（1）代表的なシグネチャファイル分割格納法であるビットスライスド・シグネチャファイル (bit sliced signature file)[7] と，（2）我々の考案した，パトリシアトライ (Patricia Trie)[9] とシグネチャファイルの組み合わせによる方式（パトリシアトライ構造化シグネチャファイル）のそれぞれを実メモリ上に作成し，全文検索コストの比較を行った．我々の実験では，ビットスライスド・シグネチャファイルは検索条件として与える文字列を増やすほど検索時間が増えるが，パトリシアトライ構造化シグネチャファイルは検索条件として与える文字列を増やすほど検索時間が減ることが分かった．すなわち，パトリシアトライ構造化シグネチャファイルの全文検索処理コストは，検索対象のテキスト数でなく，検索結果として得られるテキスト数と相関があり，多数の文書を絞り込んで少数のテキストを得るような場合に有効であるとの結論を得た．

2 シグネチャファイル

シグネチャファイルによる全文検索では，各テキストごとにテキストシグネチャと呼ばれる固定長のビット列を作成する．その作成手順は次の通り．（1）テキストの各単語ごとにハッシュ関数を適用し，固定長のビット列 (ワードシグネチャ) を生成する [8]．ワードシグネチャの長さは長大であり，そのうち一部分のビット (数ビット) のみが「1」になっている．（2）テキスト

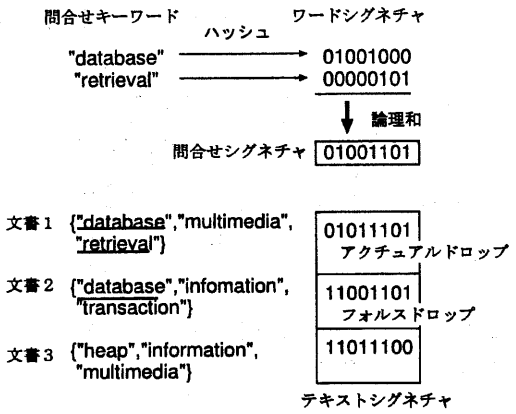


図 1: シグネチャファイルによる全文検索

に登場する全単語のワードシグネチャの論理和を，テキストシグネチャとする．

シグネチャファイルを用いた全文検索では，検索文字列の各単語にテキストシグネチャ生成と同じハッシュ関数を適応し，全単語の計算結果の論理和を問合わせシグネチャとする．テキストが検索文字列をすべて含んでいるならば，そのテキストシグネチャは，問合わせシグネチャが「1」となっているようなビットがすべて「1」になっているはずである (図 1).

シグネチャファイルによる全文検索の最も単純な実装は，全テキストシグネチャと問合わせシグネチャとのマッチングを文書の数だけ繰り返すものである．テキストの長さよりもテキストシグネチャの長さの方が短いため，単純な文字列マッチングより高速に全文検索が可能である．すなわちその走査コストは，テキスト数に比例する．

全文検索時におけるシグネチャファイル走査コストの削減のために，いくつかのシグネチャファイルの分割格納法が提案されてきた [5]．これは，シグネチャファイルを分割してディスクに格納し，全文検索時にはシグネチャファイルの必要な部分のみをディスクから読み出すようにすることで，結果として読み出されるディスクページ数を削減し，ディスクＩ／Ｏコストを削減するものである．

代表的なシグネチャファイル分割格納法であるビットスライスド・シグネチャファイルでは，各テキストシグネチャを１ビットずつに分割し，同

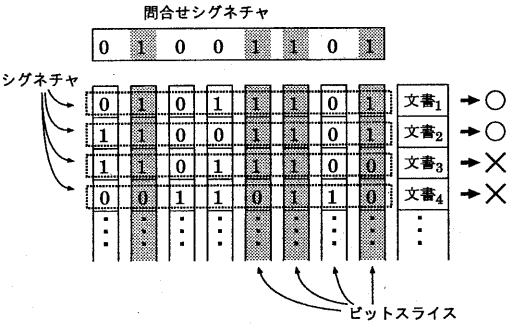


図 2: ビットスライスドシグネチャファイルによる全文検索

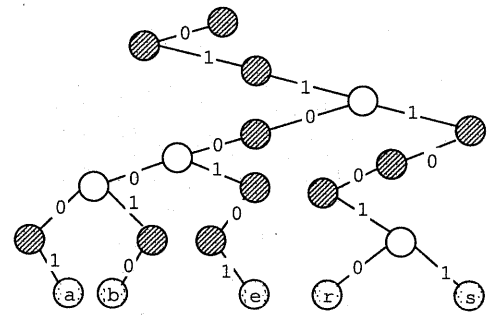
じ位置のビットをまとめて格納する．従って，全文検索では，問い合わせシグネチャのうち「1」が立っているビットに対して，テキストシグネチャの対応するビットのみを調べればよく（図 2），結果として，高速に処理できる．問い合わせシグネチャの'1' が立っているビットの数が少ないほど，調べるビットが少なくて済み，検索が速いという性質がある．

3 パトリシアトライ構造化シグネチャファイル

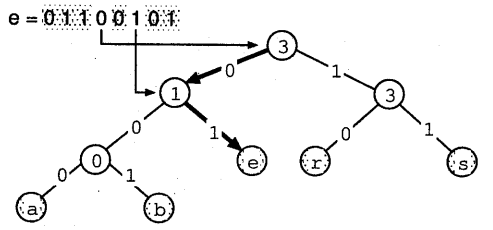
本章では，我々が [4] において提案したパトリシアトライ構造化シグネチャファイルによる全文検索法の説明を行う．

3.1 トライ

トライ (trie)[6] とは，ビットデータのための探索木であり，基本的には，ビット値が０のときは左方向に，ビット値が１のときは右方向に枝を伸ばすようなデータ構造である．例えば５つのキー “a”, “c”, “e”, “r”, “s” に対しては，各々をビットデータとみなす（例えば e = 01100101）と，図 3 (a) のようなトライを得ることができる．トライを用いた探索は，木を根から順にたどることで行う．すなわち，まず，探索キーの１番目のビットが “1” か “0” かによってたどる枝を決定し (“1” なら右，“0” なら左)，以後同様に n 段目では n 番目のビットに従ってトライをたどる．キーの全ビットに対して枝を１つ伸ばすようなトライは，フルトライ (full trie) と呼ばれ，そ



(a) フルトライ



(b) パトリシアトライ

図 3: (a) フルトライ (b) パトリシアトライ

の深さはキーのビット列の長さと同じ深さである.

フルトライに対して，節点全てが二方向分岐になるよう全ての一方向分岐の節点を 1 つの節点にまとめることで，より短くトライを格納できる．この方式はパトリシアトライ (Patricia trie)[9] と呼ばれる．図 3 (a) に示したフルトライに対応するパトリシアトライを図 3 (b) に示す．各ノードには，まとめられた枝の長さ（スキップ値）を格納する.

3.2 パトリシアトライ構造化シグネチャファイルの構成

パトリシアトライによる全文検索は S_Q をもとに $S_T \wedge S_Q \equiv S_Q$ (S_Q は問い合わせシグネチャ，S_T はテキストシグネチャ）を満たす複数のテキスト T を探索するものである．そのアルゴリズムは，問い合わせシグネチャのビットが 0 のとき枝の両側を辿り，問い合わせシグネチャのビットが 1 のとき枝の左側のみを辿るものである．但し，各節点には対応する部分シグネチャ

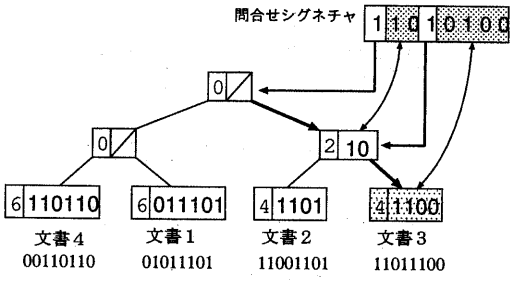


図 4: パトリシアトライ構造化シグネチャファイルによる全文検索

S'_T を格納しておき，辿りを行うごとに，訪れた接点の S'_T と，これに対応する S_Q の部分シグネチャ S'_Q について $S'_T \wedge S'_Q \equiv S'_Q$ であるかを判定し，その結果が偽ならば辿りを中止する．図 4 は，その様子を示している．根では節点内に示されたスキップ数が “0” であるので最初のビット “1” に従い右の枝のみを辿る．次の節点では，スキップ数が “2” であるので，節点内のビット列 “01” と，これに対応する問合わせシグネチャの 2,3 ビット目からなるビット列 “01” とで比較を行う．ここでは，“01” ∧ “01” ≡ “01” と条件を満たすので，次に辿る枝を 4 ビット目によって決定する．以下同様の処理を繰り返し，葉に辿りついたら，条件の確認 (例えば “1100” ∧ “0100” ≡ “0100”) を行い，終了する.

4 実験

我々は，[4] において，（1）FP (Fixed-Prefix) 分割シグネチャファイル [10] と（2）パトリシア構造化シグネチャファイルとを実メモリ上に作成し，パトリシア構造化シグネチャファイルの方が高速な全文検索が可能であることを報告した．FP 分割シグネチャファイルとは，シグネチャを先頭数ビット (prefix) のビットパターンによって分類し，それぞれをまとめてパーティション (partition) と呼ばれる単位で格納する構成法である.

本稿では，新たに（1）ビットスライスド・シグネチャファイルと，（2）パトリシアトライ構造化シグネチャファイルを実メモリ上に作成し，全文検索コストの比較を行う.

毎日新聞	16163
産経新聞	20713
讀賣新聞	36982
計	73858

表 1: 実験に利用したＨＴＭＬファイル数

文字	処理単位
ASCII 文字	単語ごと (スペース，記号で区切る)
漢字	二文字組 一文字は無視する
カタカナ	カタカナのシーケンス 一文字は無視する
数字	数字のシーケンス (ASCII コードに変換)
アルファベット	アルファベットのシーケンス (ASCII コードに変換)
平仮名	無視
記号	無視 ('ー' だけはカタカナに含める)
ギリシャ文字	無視
ロシア文字	無視

表 2: 日本語テキスト処理方式

- 実験データ

  実験には，産経新聞社様 [1]，毎日新聞社様 [2]，讀賣新聞社様 [3] の３社がWWWで公開しているＨＴＭＬフォーマットの日本語の新聞記事およそ半年分を利用した．それぞれのファイル数は表１の通りであり，ファイルサイズの平均はおよそ３Ｋバイトである．

- 実験パラメータ

  シグネチャのビット長は 1024 ビットとした．検索文字列１バイト（８ビット）につき問い合わせシグネチャの中の２個のビットを選択し１とするようなハッシュ関数を用いた．

- シグネチャ生成方式

  今回の実験では，日本語テキストを用いた．コード化方式は EUC とし，漢字については２文字組 (bigram) を単位としてワードシグネチャの生成を行った（詳しくは表２）．

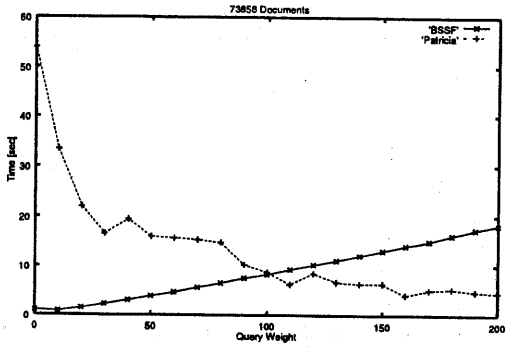


図 5: 問い合わせシグネチャのビット数と検索時間（２０００回）

- 実験方式

  実験では問合わせシグネチャの中の１のビットが立っている数をいろいろ変えることで，測定を行った．まず，問い合わせシグネチャの中のビットをランダムに選択し「１」に設定して（残りは０），全文検索と同様の処理を行うこととする．これを２０００回繰り返し，その処理時間（ＣＰＵ時間）を測定値とした．そして，問い合わせシグネチャの中のビットが「１」になっている数を０から２００まで１０きざみで変え，１本のグラフを得た．

以上の結果得られたＣＰＵ時間のグラフを図５に示す．同時に，パトリシアトライ構造化シグネチャファイルのサイズを測定したところ，１テキストあたりおよそ１５０バイトであった．

5 考察

実験結果は，問い合わせシグネチャの１ビット数が増えるほどパトリシアトライ構造化シグネチャファイルの全文検索処理にかかる時間は減少することを示している．これは，問い合わせシグネチャの１ビット数が増えるほど，検索結果として得られるテキストの数は減る（図６）ことと対応している．

今回の実験では，問い合わせシグネチャの１ビット数が１００を超えるとき（すなわち検索条件として与える文字列が日本語で２５文字を超えるとき），ビットスライスド・シグネチャファイ

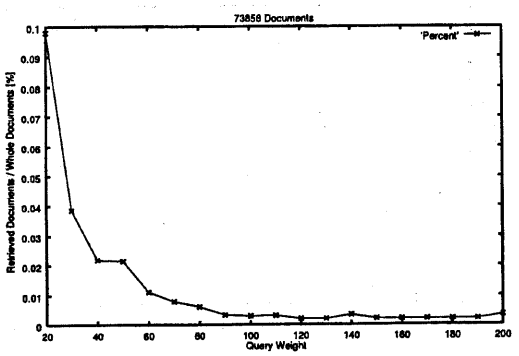


図 6: 検索結果の文書数の割合

ルよりも速いとの結果を得た．検索条件として与える文字列が日本語で２５文字を超えることは，現実にも有り得るから，本方式が有効となる局面も有り得ると思われる．

パトリシアトライ構造化シグネチャファイルのサイズは，本来のシグネチャファイルのサイズの１．２倍程度であり，テキスト１００万件あたりおよそ１５０Mバイトであることから，より大規模なテキストデータであっても十分実メモリ上に格納し得るサイズであると判断される．

6 おわりに

本稿では，問い合わせシグネチャの１であるビット数が多いとき，すなわち，検索条件として与える文字列を増やすほど，パトリシアトライ構造化シグネチャファイルの全文検索処理にかかる時間は減少することを示した．これは，ビットスライスド・シグネチャファイルとは逆の傾向である．

今後の課題は，（1）より大規模なテキストデータによる再実験，（2）今回提案した方式の解析的な評価による実用性の検討である．

謝辞

新聞記事ＨＴＭＬファイルのご提供および実験への利用のご承諾を下さった産経新聞社様，毎日新聞社様，讀賣新聞社様に感謝します．本研究は一部文部省科学研究費補助金重点領域研究（１）（課題番号０８２４４１０５），研究課題「高度応用のための情報ベースモデルとその実現技術の研究」の援助を受けている．

参考文献

[1] 産経新聞社ホームページ, http://www.sankei.co.jp/

[2] 毎日新聞社ホームページ, http://www.mainichi.co.jp/

[3] 讀賣新聞社ホームページ, http://www.yomiuri.co.jp/

[4] 権藤夏男，金子邦彦，牧之内顕文，“トライを利用したシグネチャファイルの構成法”，電子情報通信学会第８回データ工学ワークショップ，1997，pp.25-31.

[5] 渡辺悟康，北川博之，“分割ビットスライストシグネチャファイルの提案と集合値検索への適用”，情報処理学会論文誌，Vol.37，No.12，pp.2314-2325, 1996

[6] T.H.Merrett, Heping Shang and Xiaoyan Zhao, “Database Structures, Based on Tries, for Text, Spatial, and Genaral Data”, International Symposium on Cooperative Database Systems for Advanced Applications, Vol.2 December 5-7, Heian Shrine, Kyoto, pp.316-324, 1996

[7] C.Faloutsos, and R.Chan, “Fast test access methods for optical disks: Designs and Performance comparison.” in Proc. 14th Int. Conf. on VLDB. 1988. pp.280-293.

[8] C.Faloutsos, “Signature-Based Text Retrieval Method : A Survey”, Data Eng. Bulletin, Vol.13, No.1, pp.25-32, 1990

[9] D.R.Morrison. “Practical algorithm to retrieve information coded in alphanumeric”, Journal of the ACM, 15(4):514-34, 1968

[10] D.L.Lee and C.Leng, “Partitioned Signature Files: Design Issues and Performance Evaluation”, ACM Trans. Off. Inf. Syst., Vol.7, No.3, pp.158-180, April, 1989

[11] D.L.Lee and C.Leng “A Partitioned Signature File Structure For Multiattribute and Text Retrieval”, Proc, 6th ICDE, pp.389-397, 1990