細粒度全文検索法の開発

日立製作所 中央研究所
金田　泰
E-mail: kanada@crl.hitachi.co.jp

要旨: 従来のテキスト検索は，通常，文書を単位としていた．しかし，ユーザがもとめるのは文書そのものではなく，膨大な文書集合から短時間で必要な《情報》を検索することである．そこで，文書中での検索情報の存在場所へのハイパーリンクやその周辺からの抜粋が表示でき，文書中の複数の話題をくべつして全文検索できる「細粒度検索法」を開発した．この報告では，テキストを原子というこまかい単位で検索する細粒度検索のモデルを記述し，従来の全文検索エンジンをそのまま，または一部だけ改造して実現できる 2 方法を説明・比較し，試作評価によって従来法よりユーザの負荷を低減できることをしめした．

1. はじめに

従来のテキスト検索は基本的に文書を単位としている．たとえば百科事典検索のばあいは，事典項目がひとつの文書であり，それを単位として検索される．ユーザがつねに文書全体をよむのであればそれでよい．しかし，通常，ユーザがもとめているのは文書そのものではなくて文書にかかれた《情報》であり，膨大な文書集合から短時間でほしい情報をみつけることをもとめているとかんがえられる．この要求をみたすためには，つぎのような条件をみたす情報検索法を開発する必要がある．

- ユーザがもとめる情報がかかれている文書中の場所を指示することができる，かつ/または，その周辺抜粋を表示することができる．
- 文書中に複数の話題がふくまれていれば，それらをくべつして検索できる．

このような検索を細粒度検索とよぶことにする．

パッセージ検索技術はこのような条件をみたす検索技術のベースとなりうる．パッセージ検索においては文書を章，節，段落などの検索単位に分割する．そして，検索単位中の語句の出現頻度のベクトルや類似の量によってその単位を特徴づけ，検索質問にちかい特徴をもつ単位が検索結果として出力される．パッセージ検索によって文書よりこまかい単位のテキストを検索することができるが，従来，パッセージ検索はほとんど《文書》検索の性能向上のためだけにつかわれてきた．単語や文のような細粒度のテキスト単位を検索するためにパッセージ検索をつかうには，つぎのような 3 つの問題点がある．第 1 に，パッセージ検索においては統計量が重要なやくわりをはたしているので，検索単位が文のようにあまりちいさな単位になると，統計的な誤差のためにうまくはたらかない．第 2 に，テキスト単位を質問によってかえることができれば細粒度でなくても検索結果において話題を分離できるかもしれないが，パッセージ検索においてはテキスト単位を検索開始前に固定しなければならない．第 3 に，パッセージ検索においては検索単位間の関係をあつかう方法をアドホックに導入することは可能だが，そのための統一的なわくぐみはない．

そこで，上記の 2 つの要求をみたし，かつ上記のような問題点のない検索法として「細粒度検索法」を開発した．この方法には 2 つの特徴がある．

第 1 の特徴は，文書を文，語，あるいは文字というようなこまかい単位によって構成されたものとみなし，それを単位とする検索をおこなうことである．その単位を原子とよぶ．検索結果は原子内またはその周辺のテキストをふくみ，原テキスト中のその原子へのハイパーリンクをふくむ．したがって，その原子をふくむ文書中の場所や文章の抜粋をユーザに対して表示することができる．また，話題のほうが原子よりおおきければ，複数の話題を分離してあつかうことができる．

第 2 の特徴は，原子ごとに「スコア」という検索質問への適合度の評価値をつけ，原子間の関係をあらわす統一的なわくぐみとしてスコアの原子間伝播機構を導入していることである．

この報告においては，まず細粒度検索のモデルを記述し，それを実現するための 2 つの方法について説明し，それらを比較する．そして，これら 2 つの方法を実装し評価した結果をしめす．

ここで報告する細粒度検索の機能は，軸づけ検索機能の一部として Unix (Linux) 上の Perl 言語でプロトタイプを開発し，Web 上で検索できるようにした．すなわち，まず NEC において製品化された世界大百科事典 CD-ROM 版 [NEC 95] および ’95 年 1 年分の毎日新聞のテキストを使用して ’97 年に実験をおこなった．’99 年に

は，軸づけ検索機能の一部である「テーマ年表検索」(年代軸検索) が，日立デジタル平凡社において「ネットで百科」という名称の百科事典ネットワークサービスの一部として製品化されたが，そのために製品化にたえる性能および機能がえられるように Windows NT 上の Delphi 開発環境で再開発した．試作版クライアントを使用すれば細粒度検索単独で使用することができるが，製品版クライアントにおいては軸づけ検索のかたちでだけ使用することができる．

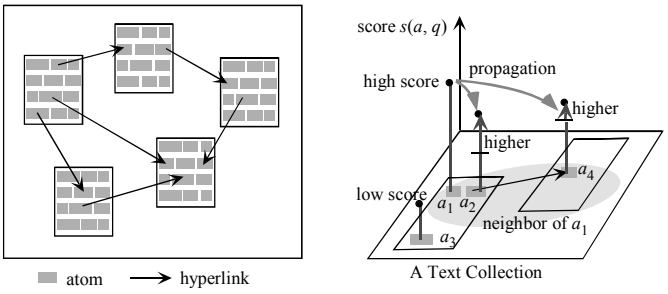
この研究の対象は大量の文字情報をふくむテキスト集合からユーザが必要な情報を検索する情報検索システムであり，電子百科事典，新聞データベース検索システム，WWW 情報検索システムなどに応用することが可能である．しかし，この研究の目標は，これまでコンピュータで自動的におこなったり支援したりすることが困難だった情報の再組織化，あるいはよりひろくいえば理解支援 [Wur 89] という，これまでの応用プログラムのわくにはおさまらない新機能を実現することである．したがって，中長期的には，コンピュータを使用した知的生産ツールのありかたを再検討すれば，ユーザ・ニーズをとらえた，あたらしいコンセプトの製品の開発につながるとかんがえられる．

2. 細粒度テキスト検索のモデル

この章では細粒度テキスト検索のモデルを，つぎの段階をおって説明する: (1) もっとも一般的なレベル，(2) 入力を文字列に限定したモデルにおける基本機能，(3) (2) のレベルにおける AND 検索，OR 検索，語彙頻度を反映した検索などの拡張機能．

2.1 一般的なモデル

より一般性があるレベルの細粒度テキスト検索のモデルについて，**図 1** をつかって説明する．このモデルにおいては，テキスト集合は文書と，文書間にはられたハイパーリンクとによって構成される (図 1(a))．文書は原子のならびである．ここで**原子**とは文字，語，文，あるいはそれよりおおきなテキスト単位のいずれかである．**ハイパーリンク**は 2 個の原子のあいだの有向辺である．かんたんにするため，ハイパーリンクの始点，終点のアンカーテキストは原子にかぎる[1]．質問 (検索要求) ごとに各原子に対して**スコア**が定義される (図 1(b))．すなわち，任意の検索要求 $q \in$ Q とテキスト集合内の任意の原子 $a \in$ A に対してスコア関数 $s(a, q)$ が定義される．



(a) テキスト集合　　(b) スコアと，値の伝播

図 1 細粒度のテキスト検索

テキスト集合は連続空間にうめこむこともできる．つまり，スコア関数の定義域 Q×A を連続空間にすることもできる．しかし，この論文においては Q, A がともに離散的である離散モデルだけについて説明し使用する．

細粒度テキスト検索の機能について，**図 2** をつかって説明する．検索操作はある閾値をこえるスコア値をもつ原子集合 $\{a_1, a_2, \ldots, a_n\}$ (あるいは原子列の集合) をもとめる．検索システムは質問 q を入力し，検索結果項目のリストを出力する．各結果項目は a_i $(i = 1, 2, \ldots, n)$ のうちのいずれかの原子に対応している．そして，結果項目はその原子がふくむテキスト，または原子をとりまくテキストのコピーをふくんでいる．このテキストが検索結果を代表し，それが有用なものであるかどうかをユーザが判断するのをたすける．結果項目はまた，もとのテキスト中にあるその原子へのハイパーリンクをもふくんでいる[2]．この点は「概念インデクス」[Woo 97] にちかい．ユーザはこのリンクをたどって原子をとりまくテキストの原文や文書全体を参照することができる．

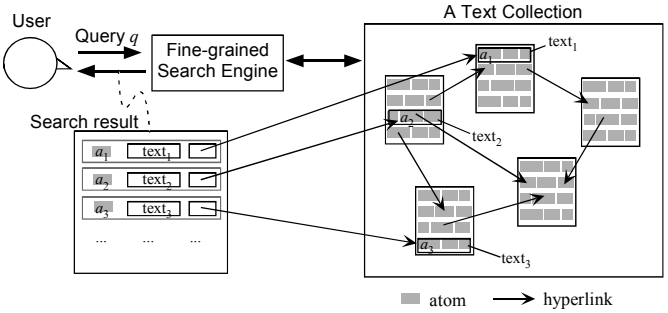


図 2 細粒度検索の機能

2.2 入力を文字列に限定したモデル

前節でのべた一般的なモデルにおいては質問の形式を限定していなかった．しかし，これ以降は，質問は，テキストに出現するべき文字列 (と，ばあいによっては AND, NOT などの演算子と) によって定義されるものとす

[1] アンカーテキストが複数の原子をふくんでいるばあいまであつかうためには，ハイパーリンクは 2 個の原子列のあいだの有効辺と定義しなければならない．もしアンカーテキストが原子の一部であるなら，この拡張をしてもまだ細粒度検索のモデルは正確にはならない．

[2] ばあいによっては，テキストからの抜粋とハイパーリンクのうちどちらか一方だけを検索結果項目にふくめることもかんがえられる．

る．すると，もし原子や原子列が指定された文字列 (あるいは指定された文字列にちかい文字列) をふくむなら，その原子のスコアはたかくなる (**図 3**)．もし原子が語や文であり，指定された検索文字列が語であるなら，原子はその文字列をふくみうる (図 3 (a))．もし原子が文字であるなら，それは複数の文字からなる文字列をふくむことはないが，その原子をふくむ原子列が検索文字列をふくむことができる (図 3 (b))．

もしある原子のスコアがたかければ，その近傍にある原子のスコアもたかくなるものとする．つまり，たかいスコア値は文書上で近傍にある原子に**伝播**される (図 1 (b) の a_2 を参照)．もし原子がハイパーリンクによって他の原子からリンクされているなら，前者は後者の近傍にあるとする．したがって，スコア値はハイパーリンクをつうじても伝播される (図 1 (b) の a_4 を参照)．たとえば，原子 a_{01}, a_{02}, … が a_0 に隣接し，$s_0(a_0, q)$ が伝播がないときのスコア関数だとすると，伝播があるときのスコア関数 $s(a_0, q)$ の計算式としてつぎのような式をつかうことができる．

$$s(a_0, q) = s_0(a_0, q) + \sum_{i>0} c_{0i}\, s(a_{0i}, q)$$

(a_0 の全隣接点についての和).

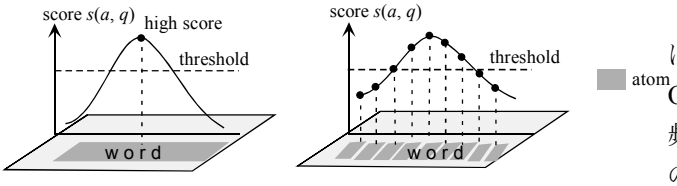
ここで c_{0i} $(0 \le c_{0i} < 1,\ i = 1, 2, \ldots)$ は定数である．この式を具体化したものとして，つぎのような式をつかうことができる．

$$s(a_0, q) = \sum d(a_0, a_i)\, s_0(a_i, q)$$

(すべての原子 $a_0, a_1, a_2, \ldots$ についての和).

ここで $d(a, a')$ $(0 \le d(a, a') \le 1$ かつ $d(a, a) = 1)$ は a と a' のあいだの距離に関する単調減少関数である．この関数は減衰関数とよぶことができる．

スコア値の伝播によるこのほかの効果については次節で説明する．



(a) ケース 1: 原子のほうがおおきいとき　(b) ケース 2: 検索文字列のほうがおおきいとき

図 3 原子と検索文字列との相対的なサイズとスコア

原子が他の原子からハイパーリンクによってむすばれているとき，前者を後者の隣接原子とみなす．したがって値は構文上の隣接点からと同様にハイパーリンクによっても伝播される．具体的な伝播の方法としてはさまざまなものがかんがえられるので，ここでは特定しない．ことなる原子からの伝播値は，基本的には加法的であるとする[1]．

スコア値の伝播によって，細粒度検索においては文書が “なめらかな” ものとみなされる．すなわち，文書中の各原子はテキストの (構文的な，または意味的な) ながれのなかにあるものとみなされる．いいかえると，細粒度検索はテキストをこまかい単位で検索するが，その単位をばらばらなものとしてではなく，隣接関係やハイパーリンク関係によって関係しあうものとしてあつかっているということができる．これに対してパッセージ検索や従来の文書検索においてはテキスト単位を基本的にはたがいに独立のものとしてあつかっている．

周囲との関係を表現するのに伝播という統一的な方法をつかっているのが，細粒度検索法の最大の特徴である．ただし，スコア値の伝播によって 2 つの問題がおこりうる．

第 1 に，細粒度検索アルゴリズムは，あるスコア値がたかい原子 a の近傍のすべての原子を出力してしまうことがありうる．そうすると検索結果は非常に冗長になる．なぜなら，近傍の原子が出力にふくまれていなくても，テキストやハイパーリンクをたどることによって，ユーザはしばしば隣接原子をみるからである．したがって，検索アルゴリズムはスコア値がたかいすべての原子を出力するのではなく，代表的な原子だけを出力することがのぞましい．

第 2 に，スコア値の伝播を文書全体，あるいはリンクされた文書集合全体というひろい範囲でおこなうと，伝播のために膨大な計算が必要になる．とくに，伝播によってえられたスコア値がさらに伝播するように伝播のモデルをきめると，収束計算が必要になって，さらに膨大な計算が必要になる．したがって，計算量が妥当な範囲にはいるように伝播のモデルをきめることが重要である．

2.3 細粒度検索における拡張検索機能

細粒度検索においては，AND, OR などの演算子を陽に導入しなくても，AND 検索，OR 検索 (AND 演算子や OR 演算子をつかった検索) にちかい検索や語彙出現頻度を考慮した検索が，スコア値伝播機構によってつぎのように実現される．

- **OR 検索のシミュレーション**

  ひとつの質問において複数の検索文字列が指定されたときは，それらのうちのすくなくともひとつをふくむ原子あるいは原子列のなかの原子のスコアがたかくなる．したがって，複数の検索文字列を指定することによって，それらに関する OR 検索がシミュレートできる．

[1] したがって，細粒度検索法は理論上は原子をニューロンとするニューラルネットとして表現するのが自然かもしれない．

- **AND 検索のシミュレーション**

  複数のことなる検索文字列がひとつの原子あるいはひとつの原子列のなかの原子にあらわれるとすると，ひとつの検索文字列だけがあらわれるばあいにくらべてスコアはたかくなる．これは前節でのべた伝播の加法性による．したがって，もし閾値をうまく制御することができれば，すべての検索文字列にちかい原子だけが検索結果にあつめられる．すなわち，複数の検索文字列の AND 検索がシミュレートできる．あるいは，スコア値の伝播の範囲を限定すれば，閾値を設定しなくても同様の効果をえることができる (その例を後述する)．この “AND 検索” においては，文書単位の AND 検索とはちがって，複数の検索文字列が一文書内のはなれた位置にあらわれてもスコアはたかくならない．したがって，たかいスコアがえられるのは，それらの検索文字列が文脈上，関連をもって出現しているばあいがおおい．

- **語彙出現頻度を反映した検索**

  ひとつの検索文字列がある原子あるいは原子列のなかの原子に複数回あらわれるとすると，1 回だけあらわれるばあいにくらべてスコアはたかくなる．これも伝播の加法性による．したがって，検索文字列の語彙出現頻度 (*tf*, term frequency) がたかい文書または文書の一部におけるスコアはたかくなる．

テキスト検索においては，検索結果の評価に語彙の出現文書数の逆数 (*idf*, inverse document frequency) を加味することがおおいが，スコア値伝播は局所的な作用であるから，細粒度検索においては，それに相当する大域的な情報はふくまれない．

AND 検索のシミュレーションについてさらに説明する．スコア伝播の方法としては，減衰関数を使用し，その値が距離と共になめらかに減少するのが自然だとかんがえられる．このばあいには，複数の検索文字列が近接してあらわれるところで閾値をこえるように閾値を設定する．たとえば，減衰関数が $\exp[-x^2/4]$ と定義され，検索文字列の数 n_s が 2 個から 4 個のあいだであり，かつ閾値が $n_s - 1$ と定義されていると仮定する．もしすべての n_s 個の文字列がおなじ原子または隣接原子中にあらわれれば (つまり距離が 0 または 1 ならば)，スコアは閾値をこえる．もし $n_s - 1$ 個の文字列がおなじ原子中にあらわれれば，スコアは $n_s - 1$ にひとしくなるが，閾値をこえない．

しかし，語彙出現頻度のような他の原因によってスコア値がたかくなることもあるので，AND 条件をみたしていないのに検索結果にふくまれることがありうる．このようなことがないクリスプな AND 検索を実現するためには，減衰関数や伝播の反復をつかわない方法をつかうことができる．この方法では，スコア値伝播をつぎのように定義する．ある原子のスコア値は，原子 1 個ぶんの距離を 1 として，そこから距離が m 以内にある原子にだけ伝播し，検索文字列をふくんでいる原子だけ実際にスコア値をたかくすると定義すれば，AND 条件をみたす原子だけをひろいだすことができる．すなわち，$\mathrm{AND}_m(s_1, s_2, \ldots, s_n)$ という質問は「検索文字列 $s_1, s_2, \ldots, s_n$ をふくむ，テキスト集合中のながさ m の原子列をもとめよ」という意味だとする．この条件をみたす原子列中の原子についてスコアがたかくなるように (伝播後の) スコア関数を定義する．この方法をとれば伝播計算もかるくすることができる．この方法において $m = 0$ とすれば，原子を文書とみなし，原子間の関係を計算にいれない，従来の論理的な AND 検索とひとしくなる．

2.4 検索例題

細粒度検索の例題として，世界大百科事典第 2 版 [HDH 98] による検索質問 AND_5(印象派，音楽) に対する結果を**表 1** にしめす．ただし，ここでは 3.2 節でのべる原子文書検索法を実装した検索システムの出力に多少の加工をくわえている．表 1 においては，項目名 (文書名) とその読み，本文抜粋 (原子がふくむテキストのコピー) のほかに，その原子をふくむ文章の見出しもあわせて表示している．本文抜粋には原文へのハイパーリンクがうめこまれている．項目名以外に見出しのないものについては，見出しの欄は空白になっている．ここでは検索結果項目数は 18 だが，文書数は 6 である．これらの結果のうち項番 2, 3, 9, 10, 11, 17, 18 は不適合だとかんがえられる[1]．

3. 2 つの実現法

細粒度検索を実現するための 2 つの方法すなわち原子位置検索法と原子文書検索法とを説明したのち，両者を比較する．

3.1 原子位置検索法

第 1 の方法「原子位置検索法」について，**図 4** をつかって説明する．この方法では各原子の位置 (address) が定義される．図 4 (b) のように原子が文字のばあいには，原子 a_i の位置は対 (d_i, l_i) によって指定される (図 4 (a) 参照)．ここで d_i はその文字をふくむ文書の識別子で

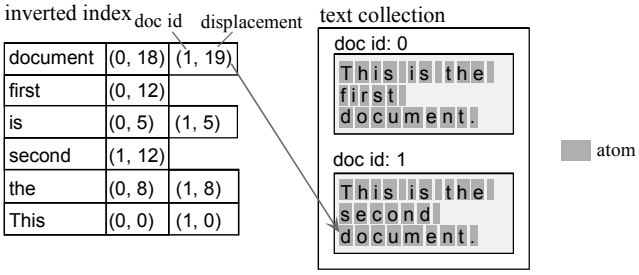
[1] ここでは仮に適合性の評価をおこなっているが，あまり客観的な評価ではない． 表 1 の項番 14 と 15 はサン・サーンスが反印象派の音楽家であることを含意しているので，印象派にかかわったものとして適合とみなした．これに対して項番 18, 19 はラロの音楽が印象派とは異質だとのべているが，印象派との積極的なかかわりをのべていないので不適合とみなした．しかし，このようなくべつは質問の意図にもかかわり，微妙である．

表 1　細粒度検索による AND_5(印象派, 音楽) の検索結果 (世界大百科事典第 2 版)

項番	項目名	読み	見出し	本文抜粋	スコア
1	印象主義	いんしょうしゅぎ Impressionnisme [フランス]		印象主義の概念は**音楽**に対しても用いられる。	0.88
2	印象主義	いんしょうしゅぎ Impressionnisme [フランス]	［起源と先駆］	部分的には，**印象派**を先取りする動きが18世紀終りごろから見られるようになる。	0.88
3	印象主義	いんしょうしゅぎ Impressionnisme [フランス]	［印象主義と日本の近代美術］	表現主義的傾向が強く，**印象派**の導入がその後の前衛絵画運動と結びついていった日本の特殊性をよく示している。	0.88
4	印象主義	いんしょうしゅぎ Impressionnisme [フランス]	【音楽】	**【音楽】**	0.99
5	印象主義	いんしょうしゅぎ Impressionnisme [フランス]	【音楽】	印象主義という概念を絵画から借りて**音楽**に適用したのは，	0.99
6	印象主義	いんしょうしゅぎ Impressionnisme [フランス]	【音楽】	**音楽**の様式について厳密に語るためには，	0.89
7	交響詩	こうきょうし		**印象派**では《牧神の午後への前奏曲》などがドビュッシーによって作られているが，	0.97
8	交響詩	こうきょうし		ロマン派的な**音楽**思潮から生まれたこのジャンルは，	0.97
9	コートールド	Samuel Courtauld		**音楽**と美術に造詣の深かった妻の影響で，	0.98
10	コートールド	Samuel Courtauld		1923年テート・ギャラリーにゴッホの《ひまわり》などを含むフランス**印象派**，	0.99
11	コートールド	Samuel Courtauld		後期**印象派**の絵画を購入する資金を寄付。	0.99
12	サン・サーンス	Charles Camille Saint-Saëns		真に独創的な**音楽**表現を生み出すにはいたらず，	0.98
13	サン・サーンス	Charles Camille Saint-Saëns		R.ビュシーヌ，フォーレ，C.フランクなどとともに国民**音楽**協会を設立(1871)，	0.98
14	サン・サーンス	Charles Camille Saint-Saëns		著述家としては反ワーグナー，反**印象派**の論陣を展開した。	0.98
15	フランス映画	フランスえいが	［フランス印象派と映画芸術運動］	映像による詩や**音楽**をつくろうとする映画芸術派が主流をなすに至った。	0.98
16	フランス映画	フランスえいが	［フランス印象派と映画芸術運動］	ジョルジュ・サドゥールが〈フランス**印象派**〉と名づけた映画作家たちが一時代をつくる。	0.98
17	ラロ	Édouard Lalo		1875)により初めて大成功をおさめ，念願の劇**音楽**では《イスの王》が喝采を博し(1888初演)，	0.98
18	ラロ	Édouard Lalo		その作風はフランクの一派や**印象派**とは異質で，	0.98

あり，l_i は文書の先頭からその文字までの変位である．変位はバイト数，文字数，あるいは他のテキスト単位の数によって計測される．テキスト全体がひとつのファイルにふくまれるばあいは文書の識別子は不要である．つまり，位置はファイル先頭からの変位だけであらわされる．

全文検索システムは語または文字の逆びきインデクスを生成し，参照する．逆びきインデクスはその語や文字のすべての出現場所をふくんでいる(図 4 (a))．逆びきインデクスは，図 4 (a) にしめしたように，原子の位置をふくむように，あるいはインデクスの内容から原子の位置を計算することができるように設計することができる．この方法においては，全文検索アルゴリズムは文書の識別子だけでなく変位をあわせてかえすようにしなければならない．



(a) 逆びきインデクス　　(b) テキスト集合

図 4「原子位置検索法」のための逆びきインデクスと原子位置の定義

従来の全文検索システムを使用して原子位置検索法を実現する技術について説明する．従来の全分検索システムは文書の識別子だけをかえす．このようなシステムをつかっても，検索結果にふくまれるすべての文書を

再走査して原子の変位をもとめることは可能ではある．しかし，この再走査は高価であり，複数の検索文字列が指定されているばあいをかんがえると煩雑である．またその結果にもとづいて計算をおこなう必要がある．この走査と計算はインデクス生成時ではなく検索時におこなわなければならないので，この方法は非常に効率がわるい．しかし，図 4 (a) にしめすように，語や文字の位置は応用プログラミング・インタフェース (API) においてはわたされないとしても，逆びきインデクスは通常，これらを保持している．したがって，原子の位置をかえすように検索システムを設計する (または改造する) ことは容易である．

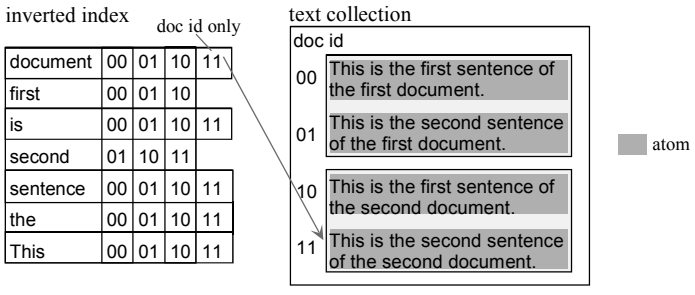
スコアリングの機構はつぎのように設計することができる．スコアリングのために減衰関数をつかうものとする．たとえば，もし原子が文字であれば，

$$d(x) = \max(0, 1 - 10^{-5}x^2)$$

という関数を，文書中の 2 個の原子間の減衰関数としてつかうことができる．ここで x は文字数でかぞえた距離である．検索時にすべての原子を評価するのは非効率なので，テキストにあらわれる検索文字列に一致する文字列の先頭位置だけを評価対象とし，そのあいだだけで伝播計算をおこなえばよい．本来はその文字列の途中でスコアが極大になりうるので，この方法でもとめられるスコアは近似値である．

3.2 原子文書検索法

第 2 の方法「原子文書検索法」について**図 5** をつかって説明する．この方法においては，各原子を文書とみなす (図 5 (b))．文書の識別子を結果としてかえす，従来の全文検索法を使用する．逆びきインデクスは原子の文書識別子のリストをふくんでいる (図 5 (a))．この方法では原子のなかに全体がふくまれる文字列だけが検索できる．したがって，原子は語かそれよりおおきくなければならない．もし原子が文字だと仮定すると，この方法ではひとつの文字だけしか検索することができないので，そのような検索システムはやくにたたない．



(a) 逆びきインデクス　　(b) テキスト集合

図 5「原子文書検索法」のための逆びきインデクスと原子位置の定義

原子のおおきさとして適当なのは文である．文がながすぎるばあいには適当に分割すればよい．文が原子であれば，それを単位とする“文書数”は本来の文書数にくらべて非常におおきくなる．したがって，検索アルゴリズムによっては深刻な性能低下がおこりうる．しかし，逆びきインデクスにもとづく全文検索システムの性能は，このばあいでも原理的にはほとんどかわらない．

スコアリング機構はつぎのように設計することができる．スコアリングのために減衰関数をつかうことができる．文書内の 2 個の原子 (ここでは文) のあいだの減衰関数の例として $d(x) = 8 / (x + 8)$ をあげることができる．ここで x は文の数によってはかった距離である．この方法においても，検索時にすべての原子を評価するのは非効率なので，検索結果にふくまれる原子だけを評価対象とし，その間だけで伝播計算をおこなう．

3.3 比較

原子位置検索法と原子文書検索法とを，いくつかの点において比較する．

- **従来の全文検索エンジンの利用可能性**

  原子文書検索法においては，文書識別子だけを結果としてかえす従来の全文検索エンジンを使用することができる．しかし，原子位置検索法においては原子の位置を知る必要があるので，効率を犠牲にしないかぎりは従来の全文検索エンジンを利用することができない．

- **テキストの変更に対する耐性**

  原子位置検索法においては，ただ 1 文字の空白が挿入しただけでも，(位置をもとめるときに空白をかぞえているならば) 空白の後方にあるすべての原子の位置がずれて，その文書に関するインデクスが無効になる．とくに ISO-2022-JP などの日本語の文字コードがつかわれているばあいには，字義がひとしい 1 バイトの文字と 2 バイトの文字とが存在する．たとえば“A”は 1 バイト文字であり，“A”は 2 バイト文字である．バイト数が一定でないため，フォーマット変換でバイト数がずれることがしばしば発生する．それにインデクスがたえるかどうかは，インデクスの保守性におおきな影響をあたえうる．これに対して，原子文書検索法はもっと耐性がある．なぜなら，文書内に出現する語彙に変化がないかぎり，それが文書内のどこにあらわれているかは検索結果に影響しないからである．

- **テキスト表示の容易さ**

  3.1 節でのべたように，細粒度検索法においては原子がふくむテキスト，または原子の周辺のテキストが検索結果項目にコピーされる．原子文書検索法において

は原子として文が適当なおおきさだが，文は表示の単位としても適当である．原子がふくむテキストをそのまま表示することは容易である．また，文は意味にまとまった単位である．これに対して，原子位置検索法においては，よりこまかい単位が原子としてつかわれる．したがって，意味のあるテキストを表示するためには原子の周辺をふくめた表示が必要であり，表示の際に適当な単位のテキストをきりだす必要がある．

以上の3点に関しては，いずれも原子文書検索法のほうが有利だとかんがえられる．それでもなお，原子位置検索法は細粒度検索を実装するときのひとつの候補である．

4. 評価

原子位置検索法と原子文書検索法とをともに実装し，評価した．この章ではまずひとつのユーザ・コストに関するモデルにもとづいて細粒度検索法と文書単位の全文検索法とを比較評価し，実装された細粒度検索法の性能を評価する．情報検索システムは，通常，適合率と再現率によって評価されるが，ここではそれらは測定していない．それは，細粒度検索法においてあるテキスト部分が適合しているかどうかを客観的にきめるのは，意味的に非常に微妙な問題をふくんでいて，むずかしいからである[1]．

4.1 文書単位の全文検索法との比較

この節では，まず細粒度検索をつかった検索のモデルと従来の全文検索をつかったモデルとを記述する．そして，細粒度検索と文書単位の全文検索における検索のモデルと，それによってユーザにかかる，コスト (ユーザの負荷 — 検索にかける時間など) のモデルをつくり，それらにもとづいて比較をおこなった．

まず，検索のモデルについてのべる．このモデルにおいてはユーザが文書全体をよまないことを仮定する．したがって，この仮定がなりたたないばあいにはこのモデルは有効でないことをことわっておく．ユーザは検索文字列を入力して検索結果をもとめ，検索対象の文書集合のなかにあらわれるすべての検索文字列をふくむ原子とその周辺だけをよむものとし，そのコストを計算する．細粒度検索においては検索結果がその原子をふくんでいる(またはリンクしている)．

文書単位の検索においては文書の先頭からテキストをなんらかの方法で走査すると仮定する．検索文字列がめだつように複数回出現するばあいがあるので，テキスト全体を走査すると仮定する．文書がみじかく検索文字列が強調表示されていれば走査は不要なので，この仮定は適合しない．しかし，文書がながければウィンドウのスクロールなどの方法によって一種の走査をする必要があるし，強調表示がないかまたはめだたないばあいには通常のテキスト走査が必要なので，これは妥当な仮定だとかんがえられる．ユーザが原子をとりまくテキストを読み，結果項目が要求をみたしているとわかれば，ユーザはさらに調査をつづけるとかんがえられる．しかし，そのコストはここでは計算から除外する．

検索コストのモデルはつぎのとおりである．検索結果にふくまれる文書数を n，そのなかでの検索文字列の出現総数を N とする (検索文字列が複数のときは，それらの出現数の総和をとる)．ひとつの検索文字列出現の周辺の文書原文テキストをユーザがよむのにかかる平均コスト (時間) を Cr とし，検索結果リストの1項目をよむのにかかるコストを Cl とする．また，文書先頭からの走査における1原子あたりの走査コストを Ct とする．

文書単位の検索においては検索結果リストの要素数は n である．したがって，それ全体をよむのにかかるコストは $Cl\,n$ である．また，よむべき原子は N 個あるので，それらをよむのにかかるコストは $Cr\,N$ である．さらに，文書 i を構成する原子数を a_i とすると原子の総数は $\sum_i a_i$ なので，走査にかかる総コストは $Ct \sum_i a_i$ である．したがって，コストの総和 Cd はつぎの式であらわされる．

$$Cd = Cl\,n + Cr\,N + Ct \sum_i a_i$$

細粒度検索においては検索結果リストの要素数は N なので，それ全体をよむのにかかるコストは $Cl\,N$ である．また，仮にユーザがつねに原子の周辺の原文テキストをよむとすると，それにかかるコストは文書検索のときと同様に $Cr\,N$ である[2]．検索文字列をみつけるのにテキストを走査する必要はないので，総コスト Cf はこの2項だけで構成される．

$$Cf = (Cl + Cr)\,N$$

文書単位の検索と細粒度検索とのコストの差 ΔC はつぎのようになる．

$$\Delta C = Cd - Cf = Ct \sum_i a_i - Cl\,(N - n)$$

ここで $Re \equiv Ct / Cl$ と定義する．つまり，原子1個ぶんのテキストを走査するのにかかるコストと検索結果リスト

[1] 2.4 節の検索例については脚注でその適合性判定が微妙で客観性がとぼしいことを指摘した．これに対して，文書単位の検索のばあいは質問とその文書の主題との適合性を判断すればよい．したがって，この例における項番 14, 15, 18, 19 はいずれも不適合とすればよく，これほど微妙な問題にはならない．すなわち，適合性判定の困難さは，文書中の複数の話題を検索しようとしたことから生じている．

[2] 実際は検索結果リスト中の1項目をよむだけでその項目が不適合だとわかるばあいがあるから，コストは $Cr\,N$ よりひくいが，ここではかんたんのため $Cr\,N$ としている．

の 1 項目をよむのにかかるコストとの比を *Re* とする．すると，細粒度検索のほうがコストがひくくなる条件つまり $\Delta C > 0$ となるための十分条件はつぎのようにあらわされる．

$$Re > (N-n) / \sum_i a_i$$

ただし，検索結果が空のときは $\sum_i a_i = 0$ となるので，この条件は定義されない．

検索対象テキストとしては CD-ROM 世界大百科事典 [HDH 98] を使用し，事典項目を文書とみなした．細粒度検索にはほぼ文を原子とする原子文書検索法を使用した．文がみじかいときは文を原子とし，文のながさが 32 バイトをこえるときにはコンマで分割している．文書数は 85,387，文数は 2,696,147，したがって 1 文書あたりの文数は 31.6 である．文書単位の全文検索としては CD-ROM 世界大百科事典に内蔵された検索エンジンを使用した．30 題の質問と，それらに関して $(N-n) / \sum_i a_i$ の値をもとめた結果を**表 2** にしめす．質問は，検索文字列を 1 個だけふくむもの 12 題，2 個を AND したもの 12 題，3 個を AND したもの 6 題で構成されている．AND した検索文字列のなかには，隣接してあらわれやすいものと，そうでないものとがふくまれている．

Re の値は検索インタフェースによって変化するが，世界大百科事典について実験したところでは *Ct* は 0.3〜0.6 秒，*Cl* は 2〜4 秒であり，*Re* は 0.1〜0.2 となる[1]．したがって，$(N-n) / \sum_i a_i$ の値はいずれにおいても *Re* よりちいさく，細粒度検索のほうが検索コストがひくいと結論できる．ただし，“AND_5(ココア，チョコレート)” のように検索文字列の出現頻度がたかい (つまり $\sum_i a_i / N$ がちいさい) 質問においては，文書単位の検索と細粒度検索とのコストの差がちぢまる．

なお，*N* が *n* にくらべていちじるしくおおきければ細粒度検索においては検索項目リストが膨大になるので不利だが，この測定結果においては *N* は *n* のたかだか 3 倍にとどまっている．

4.2 細粒度検索の実装評価

原子位置検索法および原子文書検索法をそれぞれ実装し，基本性能を測定した．いずれの実装においても細粒度検索にもとづくより高機能な検索法である「軸づけ検索」[Kan 98a] [Kan 98b] を実現しているが，ここでは細粒度検索の機能だけを評価する．また，比較のために世界大百科事典製品版にふくまれている文書単位の全文検索の性能も測定した．原子位置検索法と原子文書検索法の実装法はおおきくことなっているため，2 つの方法の比較はおこなわない．

表 2 世界大百科事典検索における $(N-n) / \sum_i a_i$ の値

質問	文書検索における結果文書数 *n*	細粒度検索における検索文字列出現総数 *N*	$\sum_i a_i$	$(N-n) / \sum_i a_i$
しからみ草紙	3	4	116	0.0086
プレスリー	6	14	860	0.0093
イリジウム	28	40	2347	0.0068
ココア	44	71	4352	0.0062
慶喜	75	129	6893	0.0078
チョコレート	84	127	11690	0.0037
浅野	223	325	16041	0.0064
コンピュータ	708	2015	100087	0.0131
生命	1386	2325	183998	0.0051
徳川	1613	2215	116928	0.0051
哲学	2146	5065	177064	0.0165
アメリカ	9785	22523	709151	0.0180
AND(ココア，チョコレート)*	7	39	375	0.0853
AND(エルビス，プレスリー)*	5	5	821	0
AND(浅野，総一郎)*	9	16	423	0.0165
AND(浅野，長矩)*	12	44	546	0.0586
AND(プレ，スリー)*	45	49	2803	0.0014
AND(徳川，慶喜)*	58	129	5512	0.0129
AND(アール，ヌーボー)*	69	132	7786	0.0081
AND(コンピュータ，通信)*	155	516	20600	0.0175
AND(生命，哲学)*	190	225	10233	0.0034
AND(徳川，家康)*	769	1337	53574	0.0106
AND(日本，アメリカ)*	4859	13315	377248	0.0224
AND(アメ，リカ)*	9807	23609	710462	0.0194
AND(浅野，総一郎，銀行)*	4	7	79	0.0380
AND(ココア，チョコレート，日本)*	4	13	128	0.0703
AND(コンピュータ，メディア，経済)*	22	3**	193	–0.0984
AND(アール，ヌーボー，絵画)*	27	30	2439	0.0012
AND(徳川，家康，秀忠)*	108	198	3560	0.0253
AND(日本，アメリカ，条約)*	527	1262	38027	0.0193

* 原子文書検索法においては AND_5 として測定した． つまり，すべての検索文字列が距離 5 以内にあらわれるばあいについてだけ *N* にカウントしている．

** 原子文書検索法においては AND_5 として測定しているため，$n > N$ となるばあいもある．

[1] ただし，検索結果リストの全体がウィンドウに表示できないときはスクロールやページめくりの時間がかかるが，ここではそれは *Cl* にふくめていない．CD-ROM 世界大百科事典においては本文中の検索文字列を赤字で表示しているが，黒字との識別がかならずしも容易でないことが *Re* の値を低下させているとかんがえられる．

原子位置検索法の実装には Perl 言語の一種である JPerl5 を使用し，逆びきインデクスには GNU DBM という不揮発性のハッシュ表を使用している．検索エンジンは JPerl5 によって記述した 2 グラム・エンジンであるが，Perl はインタプリタによって実行されるため，機械語にコンパイルするのにくらべて検索速度は 1 桁程度おそい．スコア伝播はひとつの文書全体でおこなっているが，ハイパーリンクをとおしての伝播はおこなっていない．百科事典においては文書が平均 2 kB 程度というように比較的みじかいため，文書全体でのスコア伝播によって検索時間が大幅にのびてはいない．測定結果を**表 3** にしめした．原子文書検索法の実装には Inprise Delphi 開発環境を使用しているが，検索エンジンには文書単位の全文検索とおなじ 1 グラム・エンジンを使用している．スコア伝播は，文書内で他のマッチング文字列 (検索文字列にマッチした文字列) をまたがずにあらわれるマッチング文字列のあいだだけでおこなっているため，計算量はおさえられている．測定結果は表 3 にあわせてしめした．

原子文書検索法と文書単位全文検索法とを比較する．前者は後者に対して平均で CPU 時間は 2.7 倍，経過時間は 1.2 倍かかっている．CPU 時間だけをみれば 5 倍以上かかっているばあいもあるが，経過時間をみれば，実用上，十分な性能がえられているということができる．

5. 関連研究

W. A. Woods の「概念インデクス」[Woo 97] はユーザが指定した「概念」を検索する方法である．その概念があらわれる文書中の位置へのハイパーリンクが検索される．この方法は細粒度検索の一種とみなすことができる．しかし，概念インデクス法には，伝播に相当する機構はない．

6. 結論

この研究によって，情報検索分野においてつぎのような学術的および実践的な貢献をすることができたとかんがえている．

- 細粒度検索という，あたらしい検索のモデルを提案した．細粒度検索においては検索対象のテキスト集合を原子というこまかい単位がならび，リンクされたものとしてとらえる．スコア値の伝播という機構によって，原子間の関係を統一的にあつかうことができる．

表 3 細粒度検索の 2 つの実装における検索時間 (単位: 秒)

質問	原子位置検索法		原子文書検索法**		文書単位全文検索法**	
	CPU 時間	経過時間	CPU 時間	経過時間	CPU 時間	経過時間
しからみ草紙	12.30	13	1.8	4.6	1.2	4.5
プレスリー	2.03	2	1.1	4.5	0.9	4.2
イリジウム	1.11	1	1.2	3.2	0.7	3.1
ココア	0.62	1	1.0	2.3	0.6	1.7
慶喜	0.49	1	0.7	1.4	0.1*	0.5
チョコレート	1.94	2	1.1	2.5	0.7	1.9
浅野	0.66	1	0.7	1.2	0.1*	0.8
コンピュータ	5.43	6	1.3	2.8	0.9	2.7
生命	2.49	3	0.9	1.8	0.1*	1.2
徳川	2.23	2	1.0	2.5	0.1*	0.6
哲学	5.16	5	1.1	2.1	0.1*	0.8
アメリカ	30.99	31	2.2	2.7	0.6	2.9
AND(ココア, チョコレート)****	-	-	1.9	3.8	0.9	5.0
AND(エルビス, プレスリー)****	-	-	2.0	3.9	1.1	2.5
AND(浅野, 総一郎)****	-	-	1.4	2.3	0.3	2.2
AND(浅野, 長矩)****	-	-	1.3	3.1	0.1*	1.0
AND(プレ, スリー)****	-	-	2.1	5.0	0.9	3.4
AND(徳川, 慶喜)****	-	-	1.2	1.8	0.1*	0.9
AND(アール, ヌーボー)****	-	-	2.0	2.4	1.2	2.4
AND(コンピュータ, 通信)****	-	-	1.9	3.4	0.9	3.2
AND(生命, 哲学)****	-	-	1.5	2.3	0.3	1.8
AND(徳川, 家康)****	-	-	1.4	2.1	0.1*	1.4
AND(日本, アメリカ)****	-	-	3.2	4.0	0.9	2.9
AND(アメ, リカ)****	-	-	3.2	4.0	0.7	2.4
AND(浅野, 総一郎, 銀行)****	-	-	1.5	2.5	0.4	2.5
AND(ココア, チョコレート, 日本)****	-	-	2.5	5.0	1.2	5.9
AND(コンピュータ, メディア, 経済)****	-	-	2.1	4.7	1.1	4.7
AND(アール, ヌーボー, 絵画)****	-	-	2.4	3.3	1.2	3.7
AND(徳川, 家康, 秀忠)****	-	-	1.5	2.6	0.1*	2.6
AND(日本, アメリカ, 条約)****	-	-	2.9	7.5	0.9	4.8

* 測定限界以下.

** この測定に使用したサーバは，CPU が Pentium Pro 200 MHz，主記憶が 192 MB，ハードディスクが UltraWide 7200 rpm のものである．CPU 時間，経過時間のいずれもサーバ側で測定した．

*** この測定に使用したコンピュータは，CPU が AMD K6 233 MHz，主記憶が 128 MB，ハードディスクが EIDE 5400 rpm のものである．直接，CPU 時間を測定することができなかったので，ディスク・キャッシュに当該インデクスがふくまれるばあいとふくまれないばあいの検索開始から表示開始までの時間を測定し，前者を CPU 時間，後者を経過時間としている．

**** 原子文書検索法においては AND_5 として測定した．

- 細粒度検索を実現する 2 つの方法を開発した．すなわち，原子位置検索法と原子文書検索法とである．これらの方法においては，従来の全文検索エンジンをそのまま，またはわずかな変更だけで使用することができる．これらを実装して，実用性を確認した．
- 細粒度検索をつかった検索のモデルとそのユーザ・コストのモデルをしめし，ユーザが文書中で検索文字列があらわれる部分の周辺だけをよむという仮定のもとで，細粒度検索によって従来の全文検索よりはるかに低コストで(低負荷で)検索できることをしめした．
- 文単位の細粒度検索においては，検索文字列が出現するすべての文を出力しても，文書単位の検索にくらべて，百科事典のばあいで結果項目数はほぼ 3 倍未満にとどまることがわかった．

細粒度検索を応用した「軸づけ検索」の一種である「テーマ年表検索」を日立デジタル平凡社の百科事典ネットワークサービスである「ネットで百科」において 実現した．

今後の課題についてのべる．細粒度検索によって従来の文書検索より低コストで検索ができるようになったが，インターネット，DVD-ROM などによって供給される大量のテキストに対して細粒度検索だけでは十分に対処することはできない．検索結果の組織化をともなう検索法の開発が必要である．「軸づけ検索」もそのひとつの手段だが，今後さらに，より汎用的なわくぐみや他の組織化検索法についての研究が必要である．

7. 謝辞

世界大百科事典のテキストをつかわせていただいた日立デジタル平凡社の藤井泰文取締役ほかの方々に感謝する．

参考文献

[Cut 92] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W.: Scatter/Gather: a cluster-based approach to browsing large document collections, *15th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 318–329, 1992.

[Cut 93] Cutting, D. R., Karger, D. R., and Pedersen, J. O.: Constant interaction-time scatter/gather browsing of very large document collections, *16th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 126-134, 1993.

[Goe] Goertzel, Ben: The Internet Economy as a Complex System, http://goertzel.org/ben/-ecommerce.html

[HDH 98] *CD-ROM 世界大百科事典 第 2 版* 日立デジタル平凡社, 1998.

[Kan 98a] 金田 泰: 軸づけ検索法 — 文書からの抜粋を抽出・整理して出力する 全文検索法, 情報処理学会 情報学基礎研究会報告 98-FI-50-4, pp. 25-32, 1998.

[Kan 98b] Kanada, Y.: Axis-specified Search: A New Full-text Search Method for Gathering and Structuring Excerpts, *3rd Int'l ACM Conf. on Digital Libraries*, pp. 108–117.

[Mor 95] Morohashi, M., and Takeda, K.: Information Outlining — Filling the Gap between Visualization and Navigation in Digital Libraries, *Int'l Symp. on Research, Development and Practice in Digital Libraries 1995*, pp. 151–158, Univ. of Library and Information Science, 1995.

[NEC 95] *World Encyclopædia*, NEC Home Electronics Ltd., 1993.

[Woo 97] Woods, W. A.: Conceptual Indexing: A Better Way to Organize Knowledge, *SML Technical Report*, Sun Microsystems Laboratories, 1997.

[Wur 89] リチャード・ワーマン: 情報選択の時代, 日本実業出版社, 1990.

[Zab 95] Ramin Zabih: Creating an Efficient Market on the World Wide Web (WWW からパーソナル・コンピュータなどの価格を抽出して比較できるようにした Web サイト．http://www.priceweb.com/ — 現在はアクセスできない)．Goertzel [Goe] に記述がある．