データベースシステム 99−22
（1994. 7. 21）

自己組織化マップと語彙索引を用いたデータベースの抽象化機構

銭 晴†, 史 欣‡, 田中克己‡
† 神戸大学 自然科学研究科知能科学専攻
‡ 神戸大学工学部情報知能工学科

本稿は Kohonen の自己組織化マップと語彙索引を用いたテキストデータベースの抽象化機構について述べる．この抽象化機構に基づき，文書集合の要約表示や，文書データベースの動的なスキーマ生成，文書集合の概念発掘などの機能を提供できる．また，自己組織化マップを用いて文書集合を教師なし二次元平面（マップ）上にマッピングする抽象化機構により，段階的に詳細化するブラウザーや，曖昧検索を支援するユーザインタフェースを実現でき，語彙索引を用いて提案した抽象化機構により，与えられた文書集合を表す OR で結合した問い合わせ文を生成できる．

Database Abstraction Mechanism Based on Self-Organization-Map and Word Index

Qing Qian†, Xin Shi‡, and Katsumi Tanaka‡
† Graduate School of Science and Technology, Kobe University
‡ Dept. of Computer and Systems Engineering, Kobe University

In this paper, we will describe an abstraction mechanism for text databases using Kohonen's self-organizing map and term indices. This abstraction mechanism will be useful for summarizing a collection of documents, dynamic generation of a document database schema, and helping users to discover concepts from a collection of documents. The proposed abstraction mechanism, based on Kohonen's self-organizing map, maps a collection of documents into a two-dimensional map without teaching, and it can be used for an incremental browser or an ambiguous-query interface for text databases. The latter mechanism, based on term indices, generates an disjunctive (OR) query for a given colletion of documents.

1 まえがき

テキストデータベースシステムとしては，従来のキーワード中心の情報検索 (IR) システムや，全文一致検索 (full text searching) システム，さらに関連情報をリンクでつなぎナビゲーション検索を行なうハイパーテキストシステムなどがある．しかし，これらのシステムの構築と検索は，テキストデータベースの大規模化に伴い，以下のような問題が深刻化している。

- キーワード検索とナビゲーションがそれぞれ依存しているキーワード付けやリンクオーサリングといった作業がまだ人手による作業であるため，データベースの構築・維持に大きなコストを必要とする．
- より良い検索結果を得るための問い合わせや，複雑なリンク構造を迷わず効率的にナビゲーションできるためには，ユーザがテキストデータベース全体に対する概略的な知識を持つことが必要である．しかし，現在のこれらのシステムがユーザにデータベース全体（データベースに関する知識）を見渡せるような機能とユーザインタフェースを提供していない．

このような背景から，本研究は Kohonen の自己組織化マップと語彙索引を用い，テキストデータベースの内容やその検索結果のデータ集合を要約表示したり，動的なスキーマ生成・ブラウジングを行ったりする，いわば，テキストデータベースの抽象化機構を実現する試みを行なった．本稿では，次頁の図 1に示すように，主に次の二つの事項に関して述べる．

- 文書集合からの自己組織化マップ生成とこれに基づく概覧型インタフェースの作成
- 文書集合とその語彙索引を用いた文書集合に対する質問文生成

関連研究としては、最近、津高氏 [1] が自己組織化マップをテキスト分類と検索へ適用する研究を行なっている。つまり，自動抽出したキーワードを元にテキストの特徴ベクトルを生成し，自己組織化マップのアルゴリズムによって学習させ、その結果を二次元平面上にマッピングすることにより，テキストの自動分類やキーワードマップの作成などを行なっている．本研究ではそれをさらに進め，一回で自動分類されたテキストマップ上に任意の領域を指定することによりユーザの望む内容に近いテキストを取り出す機能と，その取り出したテキストを用いて新たにマップを作ることにより，より詳細な分類情報を取得する上で、ユーザに望まれるテキストに近付いていく機能を実現した．

また、予め作った語彙索引を用い、与えられた文書集合を表せる質問文、つまり、いくつかの「あるキーワードを持つテキスト」という最もシンプルな問い合わせ Q_i が **OR** で結合した質問文 $(Q_1 \cup Q_2 \cup \ldots)$, の自動生成機構を実現した。この機構で生成した質問文により、文書集合にある主な概念を発見することができる。

これらの抽象化機構により，a. テキストデータベース全体を概略から詳細へと段階的にブラウジングする機能を備える問い合わせユーザインタフェースや，また，b. 概念獲得によりハイパーリンクを作成する支援ツールなどを実現することが期待できる．

2 自己組織化マップを用いた抽象化機構

2.1 自己組織化マップ

ニューラルネットワークの一種である自己組織化マップ（Self-Organizing Map、以下に SOM 法と呼ぶ）は，T. Kohonen により提案された中間層のない 2 層型の教師なし競合学習モデルである [2]．このモデルの特徴はデータに隠されているトポロジカルな構造を学習アルゴリズムにより発見し，通常 2 次元空間で表示するということである．

具体的には，入力データを通常高次元の特徴 (feature) ベクトル x にパターン化し，出力層にある各ユニット i が入力パターン x と同次元のベクトル m_i を持っており，2 次元平面上に配置される。学習はこれらのユニットが入力パターンに選択的に近付けることによって進行する．競合というのは SOM 法が入力パターンに一番近いパターンを持つ出力ユニット c 及びその近隣のユニットの集合 N_c のみが入力パターンに近付けらせるようなアルゴリズムを取っている．また，統計的に正確な学習効果を得るため，一定の学習回数 T を取らなければならない．SOM 法のアルゴリズムは以下のとおりである．

1. 各入力データをパターン化する。

   $X = \{x_1, x_2, \ldots, x_m\}$.

2. 出力層にある各ユニットの持つパターンを初期化する。

   $M = \{m_1, m_2, \ldots, m_n\}$.

3. 入力パターン x_k に対して、x_k と一番近いパターンを持つ出力ユニット c を探す。

$$\|x_k - m_c(t)\| = \min_{\text{for all } i} \{\|x_k - m_i(t)\|\}$$

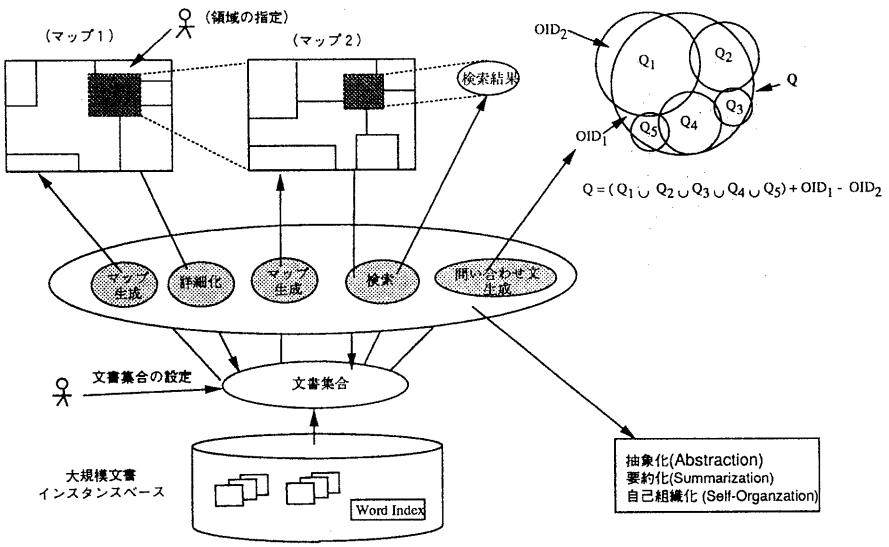

図 1: テキストデータベースの抽象化機構

$\|\cdot\|$ は距離を表し、Euclidean 距離等が用いられる。

4. **出力ユニット c とその近隣のユニット集合 $N_c(t)$ を入力パターン x_k に近付ける。**

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)[x_k(t) - m_i(t)], & (i \in N_c(t)), \\ m_i(t), & (i \notin N_c(t)). \end{cases}$$

$$\alpha(t) = \alpha_0(t) exp(-\|r_c - r_i\|^2/\sigma(t)^2)$$

ここで$\alpha(t)$ は学習率であり時間とともに 0 へと単調減少する。$r_c - r_i$がユニット c と i との距離であり、図 2 の様に取る。$N_c(t)$ の大きさも時間とともに単調に減少する。$\alpha_0(t)$、$\sigma(t)$ としては単調減少の一次関数や指数関数がよく用いられる。

5. $k = k+1, k \leq m$, 3～4 を繰り返す。

6. $t = t+1, t \leq T$, (T が予め設定された学習回数であり)、$N_c(t)$ とα を次第に小さくしながら 3～5 を繰り返す。

以上のアルゴリズムで学習した結果は、入力ベクトル空間で近くにあるものが、ネットワーク上でも互いに近傍のユニットへと射影されるような写像が完成することになる。

2.2 テキストの自動分類と検索への試み

2.2.1 テキストの特徴ベクトル化

テキストの特徴となる単語が自動的にテキストから切りだされ、単語の種類を次元とし、各要素が単語の出現頻度に比例するようなベクトル表現を用いることによってテキストをパターン化する。以下にこれをテキストベクトルと呼ぶ。例えば、(stop words を除く)

"The more I get, the more I want."

↓

textvector("more" → 2, "get" → 1, "want" → 1)

↓

正規化する

2.2.2 学習アルゴリズム

上でベクトル化されたテキストを入力データとして学習させる。学習アルゴリズムは 2.1 節で述べたものであり、その中のいくつかパラメターの設定が重要である。

- 出力ユニットの初期値

  マップ上の出力ユニットの初期値はすべて 0 とした。

- マップの大きさ

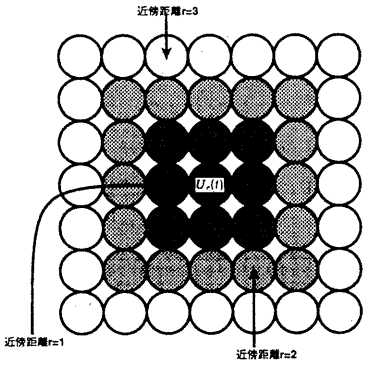

図 2: 近傍の取り方 (一例)

マップの大きさは直接に各々の テキストベクトルの次元と関係があり、f(ベクトルの次元) のような関数で指標として与えるのが妥当であるが、 関数作成に対する明確な目安がないため、マップのインスタンスを生成する際に指定するように設計した。

- 学習率α

学習率は 2.1 節で述べた数式をそのまま使う。その中の$\alpha_0(t)$ と$\sigma(t)$ が次のようにする。

$$\alpha_0(t) = 0.5 \times (1 - \frac{t}{\text{学習回数}})$$

$$\sigma(t) = (1 - \frac{t}{\text{学習回数}}) \times (\text{マップの長さの}\frac{1}{2})$$

- 学習回数t

学習回数は、テキストベクトルを学習させる結果マップの品質に大きな影響を及ぼす。2.1 節に並べている数学式を見てみると分かるように、学習回数が少ないと個々の入力パターンの出力ユニットに対する影響力において学習回数に依存した格差が生じ、学習の初期においてなされた最適でない学習の効果を補正しきれない危険が出てくる。しかし、最適な学習回数を決めにくいので、マップの大きさと同様に、ユーザの調整できるように設け、実験を行って決めることにした。

- マップ構造のループ化

マップの上下左右を連続にすることによって、マップ上の領域のゆがみの防止とマップのスクロール機能を実現する。

学習後、テキストを分類する。つまり、各々のテキストを再びマップ上に入力し、もっとも近いパターンを持つ出力ユニットへマッピングされる。

2.2.3 マップ表示アルゴリズム

学習したマップを見やすく表示するために、距離の近いベクトルを持つユニットが同じ領域に、距離のちょっと離れているベクトルを持つユニットが違う領域に分割できるように、境界線を引くことにした。距離の遠近の判断はユーザの指定によるものであり、指定された数字より小さいのは近いといういみであり、大きいのは遠いといういみである。また、分割された領域に一番強い特徴、つまり最も出現頻度が高いキーワードを選び、その領域のラベルとして付けられる。

2.2.4 曖昧検索

マップを導入した情報検索の形式としては、点指定検索と領域指定検索が設けられる。点指定検索というのはマップ上の任意点をマウスで指定すると、その一点と対応しているユニットにマッピングされたテキストが別のウインドウに出力される。領域指定検索というのはマップ上の任意の領域をマウスで指定すると、指定された領域に含まれた全てのユニットにマッピングされたテキストが別のウインドウに出力される。その二つの検索形式はいずれもユーザーにキーワードの入力を要求しないものであり、ユーザが欲しい情報を求めるため、適当な位置を指定する形で、目的のデータに近付くというアプローチを取り、より使いやすいユーザインタフェースを提供している。

もっと特徴になるのは曖昧検索できるということである。SOM 法による生成したマップが、図 3に示すように、同じ領域に表示されるキーワードが同じでも、マップ上の位置が異なるとマッピングされるテキスト内容も異なる。自己組織化マップのこの性質を活用し、普通の問い合わせで表現しにくい曖昧検索もできる。例えば、(図 3を参照)、'タイヤの構造' に関するテキストを求める場合には、'タイヤ' という領域と'構造' という領域の間のところを指定して検索すればよい。'エンジンの構造' に関するテキストを求める場合には、'エンジン' という領域と ' 構造 ' という領域の間のところを指定して検索すればよい。また、'一般的な構造論' に関するテキストは'構造' という領域の真中のところを指定して検索すればよい。

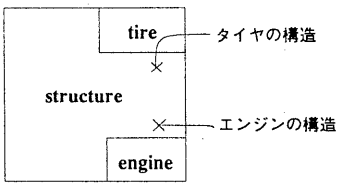

図 3: マップ上の曖昧検索

また、テキスト A と B の間の内容を持つテキストが欲しいというような曖昧検索もできる。つまり、テキスト A と B をまずベクトルし、マップ上の最も近いベクトルを持つユニットに反応させる。反応された二つのユニットの中間点を指定してアクセスすればよい。

2.2.5 マップの段階的詳細化 (zooming)

マップの詳細化 (zooming) というのは二つの意味がある。

- マップのサイズを拡大し、領域を細かく分割できるように境界線を描画し、もとのマップ上に表示できなかった領域やキーワードを表示できるようにする。この意味の詳細化 (zooming) は通常マップ全体に対する行なわれる作業である。
- ユーザがマップの一部である領域を指定し、その領域にマッピングされた（一定の内容範囲の）テキストのみを用いて新たにマップを作り直す機能である。この機能により、最初に（マップのサイズの限定で）大雑把に分類しかできなかったテキストに対し、細かく分類でき、ユーザが段階的に自分望む内容のテキストに近付いていく。この意味の詳細化 (zooming) はマップ上の一部である領域を指定する必要がある。その方法は以下の 2 種類がある：

  1. **キーワードで予め分割された領域の指定**マップに表示されたキーワード 1 個または複数個をマウスで指定し、そ（それら）のキーワードでラベルされている領域に分類されるテキストのみを用いて新たにマップを作り直す。単一のキーワードを指定した場合、そのキーワードを無視してマップを作り直す。
  2. **マウスで任意領域の指定**
     マップ上の任意の矩形領域をマウスで指定し、その領域に含まれるテキストのみを用いて新たにマップを作り直す。

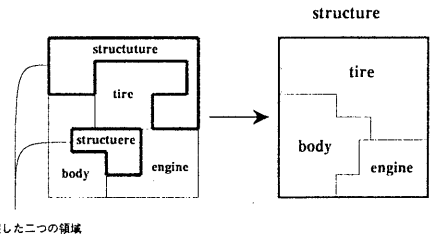

図 4: 分裂した領域の形を zooming により補正

またキーワードを指定して zooming することにより、ゆがんだマップを補正できる。学習時、テキストベクトルがマップに影響を与える順序はランダムであるため、矩形に広がるはずの領域が他の領域の影響で形をゆがめられたり複数の領域に分裂してしまう時がある (図 4参照)。このようなときキーワード指定の zooming を行うと、ゆがんだり分裂したりした領域が矩形の枠の形に整えられて再表示されることになる。この意味で zooming 機能がマップ表示の補足作業としても使われる。

2.3 テスト例とその結果の分析

今回行なったテスト例には、学習させる入力データとして、OODB 関連のテキスト 100 件のタイトル を 2.2.1 節に述べた方式でパターン化したものにした。学習式は 2.1 節と 2.2.2 節で述べた式をそのまま用いた。また、現在の実現段階で学習が遅いため、学習回数がかなり少ない 100 回に、マップサイズも小さく 8×8 にした。結果を図 5に示している。

この結果を分析してみると、次のようなことがわかった。

- テキストの分布状況

  100 件のテキストがマップの一箇所に集中せずに全体に分布していることがわかった。また、多くのテキストに見られる'language', 'programs' などの単語が大きな領域を占めており、マッピングされたテキスト量と領域面積の関係が正しく、見通しの良いマップができる。

- マップ品質と学習パラメータ

  今回のテストでは、マッピングされたテキスト間の距離でこれらのテキストの内容関連度を正しく反映するマップを得たとは言い難しかった。それは 2.2.2 節で分析した学習パラメータの不適当や、特に学習回数

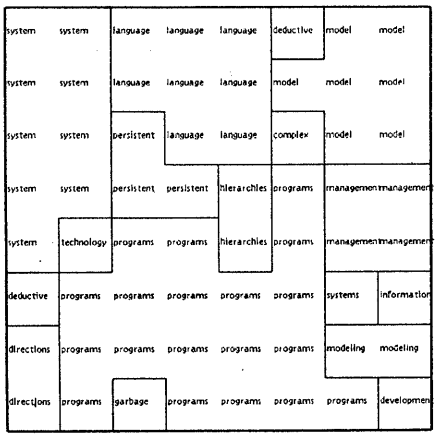

図 5: OODB 関連の文献を分類したマップ

の少ないすぎるなどが主な原因になっている。この意味で、かなり速く学習できるような実現でないと、自己組織化マップ SOM 法がまだ応用できてないと言える。特に本研究の特徴と言える zooming 機能がマップ再計算する必要があり、対話的な利用に耐えるような実現速度が望まれるため、学習速度の改善が非常に重要である。

- キーワードマップ

  テキストが自動抽出したキーワードにより特徴パターン化されるため、SOM 法によりパターン化したテキストを分類した結果マップが本質にキーワードに基づいて得たマップとも言える。このテストの結果を見るとわかるように、多くの一般的な言葉（例えば，'language'、'system' など）がマップ上によく出現し，これらの言葉が意味的に特定できないため，それに基づいてテキストを分類することも意味がないものである．意味をはっきりしているマップを得るため，テキストからキーワードを抽出する時，文脈を考慮する必要がある．例えば，'system' という言葉を抽出するより，文脈を付いている'object-oriented database system' や'multimedia information system' などを抽出したほうがいい．

利点としては、現状でも次の二つが考えられる。

- 各領域のマップ上での広さをみることによってその領域に分類されているテキストの分量をテキスト全体に対する相対的な量として知ることができる見通しの良さ
- キーワードを用意する必要がない検索作業の容易さ

3 語彙索引を用いた質問文の生成機構

3.1 概論

テキスト集合から質問文を生成する機構を論じる前に、まず生成しようとする質問文（問い合わせ文）の構成について限定をしてあることを述べておく。本研究においては、問い合わせ文として、**「単語〜を持つもの」という問い合わせの OR による結合** だけを考える。（つまり、「〜の後に〜のあるもの」や「〜を〜個以上持つもの」などの問い、あるいは AND, NOT による合成は考慮に入れない、ということである。）

ここでの問題は、あるテキスト集合 Sを内包的に表現できる問い合わせ Q が以下のような理想的な形式でないものが多い。($|\,Q\,|$ が Q で検索した文書集合を表す)

$$|\,Q\,| = |\,q(t_1) \cup q(t_2) \cup \ldots \cup q(t_n)\,| =$$

$$D(t_1) \cup D(t_2) \cup \ldots \cup D(t_n) = S$$

つまり、完全に内包的な問い合わせ Q で Sを表現する場合、過不足表現や余分表現などが多い：

$$(|\,Q\,| - S) \subset S \quad and \quad |\,Q\,| \supset S$$

結局、集合 Sが内包的な集合表現（問い合わせ $Q = \cup q(t_i)$）のみによっては表現できないため、外延的な表現（テキストオブジェクトの oid 表現）により補完し、次のように表現せざるを得ない。

$$S = |\,Q\,| \cup O^+ - O^-$$

その中の Q と O^+と O^-が次のようである。

$$|\,Q\,| = |\,\cup q(t_i)\,| = D(t_1) \cup D(t_2) \cup \ldots$$

$$O^+ = S - |\,Q\,|$$

$$O^- = |\,Q\,| - S$$

集合 Sを表現する問い合わせ Q がいくつかの組合せがあるため、その中に最適な Q を選択する尺度が必要であり、それについて次に説明する。

3.2 評価尺度

本研究では、ある問い合わせQで集合Sを表現して適当かどうかを図るために、情報検索システム (IR) によく使われる再現率 (recall factor) と適合率 (precision factor) を用いることにした：

$$\text{再現率} = \frac{|S \cap |Q||}{|S|}$$

$$\text{適合率} = \frac{|S \cap |Q||}{||Q||}$$

この二つの尺度が両方とも0から1までの数値であり、1に近い方が望ましいが、一方が高ければ他方は低いという相反する関係がある。二つ相反する尺度を統合して一つ評価尺度にするため、次のような重みづけをした評価関数を用いた。

$$f_s(Q) = \alpha\frac{|S \cap |Q||}{|S|} + \beta\frac{|S \cap |Q||}{||Q||}$$

$$(\text{ただし、}\alpha+\beta=1, \alpha \geq 0, \beta \geq 0)$$

3.3 質問文生成アルゴリズム

本研究で取っている質問文生成アルゴリズムはバックトラック (backtrack) 法により、与えられた集合Sを表現する問い合わせ文の最適な組合せを探索するものである。q_iを「単語～を持つもの」という問い合わせ文であるとし、$q_1, q_2, \ldots, q_n$の組合せによりなるべく外延的表現を用いずにSを表現することを考える。

以下に、そのアルゴリズムを説明する。(すでにSを$q_1, q_2, \ldots, q_{n-1}$までは質問文にできているとし、当段階は、$q_n$を決定するものと想定する。)

1. S'中に含まれる全ての単語$t_1, t_2, \ldots$を得て候補とする。(このとき、$S' = S - |\cup_{i=1}^{n-1} q_i|$とする)
2. 次の条件を満たさない単語t_iは候補から除外する。(あらかじめp, q, rを定めておく)
   (a) $|S' \cap D(t_i)| > p$
   (S内部に新たに含むことになるデータ数)
   (b) $|\overline{S} \cap (D(t_i) - |\cup_1^{n-1} q_i|)| < q$
   (S外部に新たに含むことになるデータ数)
   (c) $f_{S'}(D(t_i) - |\cup_1^{n-1} q_i|) > r$
   (問い合わせの逓増分についての評価関数)
   (d) $f_{S'}(D(t_i) - |\cup_1^{n-1} q_i|)$を最大にするものを$t_m$としたとき, $(0 < w \leq 1)$
   $f_{S'}(t_i - |\cup_1^{n-1} q_i|) \geq w f_{S'}(t_m - |\cup_1^{n-1} q_i|)$
3. 候補中で、$q_n = t_i$としたときの組合せ$q_1, q_2, \ldots, q_n$がすでに試されている場合は除外する。
4. 一つも候補 (となる単語) が残っていない場合にはこの組合せの探索を打ち切る。
5. 候補として残ったものを、2.(c) で用いた$f_{S'}(D(t_i) - |\cup_1^{n-1} q_i|)$の大きさにより整列すると, $t_1, t_2, t_3, \ldots, t_j$となるとする。そのとき, $S - D(t_1), S - D(t_2), S - D(t_3), \ldots$の順に深さ優先探索をしていく。

$q_1, q_2, q_3, \ldots$と処理が進展するにつれ、被覆する領域は$q_1, q_1 \cup q_2, q_1 \cup q_2 \cup q_3, \ldots$の様に広がっていくが、同時に、それぞれの時点で$S$は次のように表される。

$$S = |q_1| + O_1 - O_1'$$

$$S = |q_1 \cup q_2| + (O_1 \cap O_2) - (O_1' \cup O_2')$$

$$\vdots$$

$$S = |\cup_1^n q_i| + \cap_1^n O_i - \cup_1^n O_i'$$

ただし、

$$O_i = S - (S \cap |q_i|) \quad and \quad O_i' = q_i - (S \cap |q_i|)$$

であるとする。O_i, O_i'はそれぞれSの内部、外部においてoid表現を要するデータ集合である。

こうして得られた組合せが幾つかあり、その中に最適なものを判断し選択するのは次のような尺度を取っている。

1. $f_s(\cup_1^n q_i)$が大きい
2. $|\cap_1^n O_i| + |\cup_1^n O_i'|$が小さい
3. nが小さい

これらのうち、どれも重要な尺度であるが、1～3を特定の順序で用いることにより、最終的な問い合わせ文となる$\cup_1^n q_i$を決定する。ちなみに、本研究中の事例においては、1. により候補数を減らした後で、さらに、3. を用い冗長な表現を除く、といった順序をとっている。しかし、最終的には単語の意味が鍵となるため、いくつかの選択肢から一つの組合せを決定するのはユーザーの判断に任すことにしている。

このアルゴリズムの有効性を証明する実例は図6に示される. 入力データがOODB文献から集まった41件テキスト集合であり, 質問を構成する代表的なキーワードが階層構造に置かれ, 各々のキーワードがその上に置かれてい

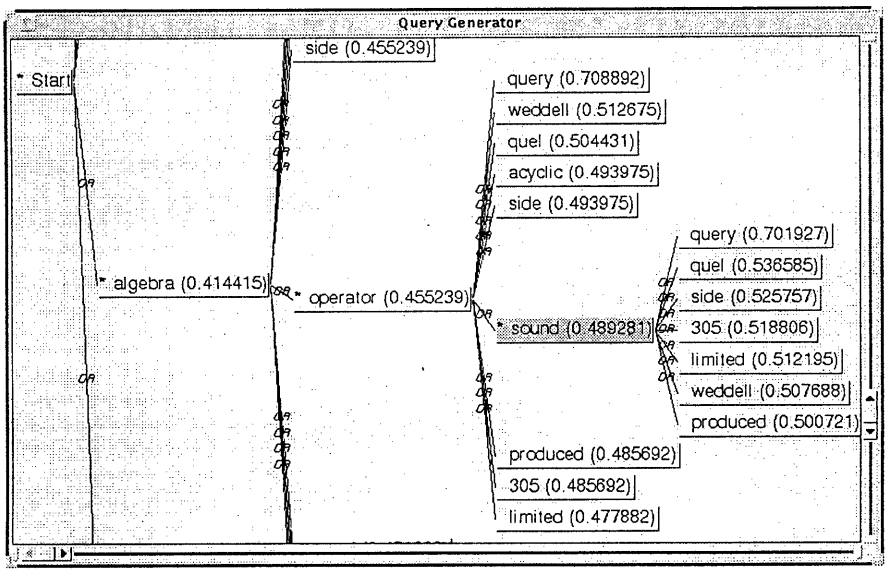

図 6: 質問文生成の実例

る全てのスーパーキーワードと OR で結合した質問文の評価値も示される。評価値の高い結合が勝ちとしたら，図 6 から分かるように生成した質問文は次のようである:

'algebra' OR 'operator' OR 'sound' OR 'query'.

4 今後の課題

- 大規模データベースの抽象化に対応できるため，自己組織化マップのアルゴリズムの簡単化
  - 特徴ベクトル間の距離を測るためには，ハミング距離を用いる [5].
- 文書データベースの抽象化構造を表す正確なマップを得る為には，context を考え入れた文書パターンの自動生成機構の導入.
- 複雑な構造を表せるためには，階層構造を取る自己組織化マップの導入.
- 領域をきれいに閉じれるよう，即ちマップ調整 (fine tuning) アルゴリズムの導入.

謝辞

本稿で示したソフトウエアの開発に参画いただいた神戸大学工学部計測工学科の服部元宏氏と大磯洋明氏に深く感謝の意を表します。

参考文献

[1] 津高新一郎「自己組織化マップを用いたテキスト分類の試み」, 情報処理学会第 4 6 回全国大会, 分冊 4 , pp.187-188, 1993

[2] Kohonen,T., *The Self-Organizing Map*, Proceedings Of The IEEE, Vol.78, No.9, pp.1464-1480, 1990

[3] 服部元宏,「自己組織化マップを用いた情報検索に関する研究」, 神戸大学工学部計測工学科卒業研究報告, １９９４年３月

[4] 大磯洋明,「サイエンテイフィックデータベースにおけるアクテイブルール機構に関する研究」, 神戸大学工学部計測工学科卒業研究報告, １９９４年３月

[5] Abramson,N. 著, 宮川洋 訳, 情報理論入門, 好学社, 1971