

意味の数学モデルによる意味的連想検索の 高速化アルゴリズムとその実現方式

宮原 隆行[†] 清木 康^{††} 北川 高嗣[†]

データベースを対象としたデータ検索における基本操作は、連想検索である。本論文では、パターン・マッチングではなく、意味的な等価性、類似性に関する計算によりデータベース検索を行う意味的連想検索方式の高速化アルゴリズムを提案する。本アルゴリズムは、意味的連想検索方式として、意味の数学モデルを適用した方式を対象とする。意味の数学モデルによる意味的連想検索の特徴は、文脈や状況に応じた意味の扱いを実現する点にある。本アルゴリズムは、キーワード、および、それを説明する文脈語列を受けとることによって意味的に関連する情報を高速に抽出するために用いられる。このアルゴリズムの実現可能性、および、有効性を明らかにするために、基本英単語を対象とした意味的連想検索の実験、および、実在するデータベースを対象とした意味的連想検索の実験を行った。それらの実験結果により、提案アルゴリズムが意味的連想検索の高速化の実現に有効であることを明らかにする。

A Fast Algorithm and Its Implementation Method for Semantic Associative Search by a Mathematical Model of Meaning

TAKAYUKI MIYAHARA,[†] YASUSHI KIYOKI^{††} and TAKASHI KITAGAWA[†]

The basic operation for extracting information from databases is associative search. In this paper, we present a fast algorithm of semantic associative search which realizes information extraction by the computation on semantic equivalence and similarity. This algorithm is applied within the framework of the semantic associative search method which has been proposed as a mathematical model of meaning. The most important feature of this method is that it provides a machinery for computing contexts and situations. This algorithm is used to extract semantically related information by receiving a keyword and a sequence of the context data items which explains the keyword. To clarify the feasibility and effectiveness of the algorithm, we have performed several experiments in which the fast algorithm is applied to semantic associative search for the basic English word retrieval and the actual database search. Those experimental results show that the fast algorithm is effectively used to extract the potential ability of the semantic associative search method.

1. 序 論

1.1 研究の背景

データベース・システムにおけるデータ探索のための主要な基本操作は連想検索である。ここで、連想検索とは、あるキーワードに関連する情報をそのキーワードが表すアドレスではなく、そのキーワードの内容に応じて検索することをいう^{10),15)}。現行のデータ

ベース・システムにおける連想検索は、パターン・マッチングによる検索であり、異なる表現形態であるが同一の意味を持つデータや近い意味を持つデータの検索を行うことはできない⁹⁾。また、同一のデータが持つ多義性を取り扱うことはできない。データ間の意味的な関係の扱いについては、データ間の関係を静的かつ明示的に記述し、同一性、相異性を判定する方法が広く用いられてきた^{2),12)}。しかし、その判定は、静的に与えられた関係を用いて、曖昧性を含んで行われる。たとえば、シソーラスを用いて同義語を照会する方法があるが、その同義語は、シソーラスの設計時に静的に決定され、同義であることの定義には曖昧性を含んでいる。また、各単語について、その意味を表すベク

[†] 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba

^{††} 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University

トルを形成し、ベクトル間の相関の強さによって意味の近さを判定する方式がある³⁾。その方式においては、ある単語に意味的に近い単語を求めるために、他の全単語各々に対応するベクトルとの間で相関の強さを計算しなければならない。

我々は、データ間の意味的な同一性、相異性は、静的な関係によって決定されるのではなく、文脈や状況に応じて動的に変化するものであり、その動的な要素を含んで決定しなければ、データ間の関係の曖昧性を排除することはできないものと考えている。このようなデータ間の意味的な関係を文脈に応じて動的に計算するモデルとして、意味の数学モデルが提案されている^{6),7)}。

意味の数学モデルは、ほぼ無限通りの文脈や状況に応じて動的に変化するデータ間の意味的な関係を計算することを目的としたモデルである。このモデルに基づいた連想検索により、文脈に応じて、検索キーワードに意味的に近い検索対象語を連想検索することが可能となる。検索対象となるデータが大量となった場合には、意味の数学モデルによる意味的連想検索の高速化が必要になる。意味の数学モデルは、ある単語に意味的に近い単語を、大量のデータの中から高速に抽出する能力を潜在的に備えている。これは、このモデルにおいて、意味の近い2つの単語は、空間上における距離の近い2点として配置されていることによる。

本論文では、意味の数学モデルによる意味的連想検索において、文脈に応じた高速な連想検索を行うためのアルゴリズムを提案し、その実現方式を示す。

パターン認識、計算幾何学の分野においては、比較対象あるいは検索対象のデータ集合の各要素が固定的な位置にあることを利用してポロノイ図をあらかじめ計算し、ポロノイ図を利用して高速化を実現する方法が提案されている^{9),10)}。また、空間データベース (spatial database) の分野においては、検索対象となるデータ (オブジェクト) が固定的な位置に存在することを利用し、それらのデータのグルーピング (クラスタリング) を行い、インデックスを作成することにより、検索の高速化を行っている^{4),13)}。

これらの高速化方式との比較において、本連想検索方式の高速化の実現では、動的に変化するデータ間の位置関係を前提としたアルゴリズムの設計が必要となる。本連想検索方式では、正規直交空間上に、検索者が発行する検索キーワード (検索語ベクトル) および文脈語に対応するベクトル、および、検索対象 (比較対象) データに対応するベクトル (比較対象語ベクトル) を写像する。

そして、検索キーワード、および、その文脈を確定

するための文脈語群が与えられると、その文脈に対応する正規直交空間の部分空間を動的に選択する。その部分空間における検索語ベクトルと比較対象語ベクトルの位置により、その部分空間において検索キーワードと意味的に最も近い比較対象データを動的に抽出する。このように、本連想検索方式では、文脈理解を正規直交空間における部分空間選択によって実現し、与えられた文脈、すなわち、選ばれた部分空間において検索キーワードに最も近い比較対象データを抽出する。

この連想検索機構では、文脈に対応する部分空間上において、検索キーワードと比較対象語の位置的な近さを計算するので、各検索ごとに、対応するベクトル群を、選択された部分空間上に動的に写像する。すなわち、検索ごとに動的にベクトル群の位置関係が決まるので、検索の高速化を実現するために静的なインデックスをあらかじめ作成しておくことはできない。そこで、本連想検索においては、文脈に応じて動的に変化するデータ間の関係を前提としたアルゴリズムを実現する。

本方式は、多変量解析による空間生成を用いた情報検索手法¹⁶⁾とは、次の点で本質的に異なる。本方式では、直交空間における部分空間の選択を行う演算 (意味射影) を定義し、その演算により、言葉の意味を文脈に応じて、曖昧性を排除して解釈する機構を実現している。この機構により、検索キーワードと検索対象データの間の意味的な関係を、与えられた文脈に応じて動的に計算することを可能としている。

2. 意味の数学モデルによる意味的連想処理

2.1 概要

ここでは、対象としている意味の数学モデルの概要を示す。

- (1) 前提: いくつかの単語を特徴づけたデータの集合が、 m 行 n 列の行列 (以下、“データ行列”と呼ぶ) の形で与えられているものとする。この行列において、 m 個のそれぞれの単語 (word) は、 n 個の特徴 (features) によって特徴づけられている。
- (2) イメージ空間 I の設定: データ行列から、特徴づけに関する相関行列をつくる。そして、相関行列を固有値分解し、固有ベクトルを正規化する。相関行列の対称性から、得られたすべての固有値は実数であり、その固有ベクトルは互いに直交している。このとき、非ゼロ固有値に対応する固有ベクトル (以下、“意味素”と呼ぶ) が形成する正規直交空間をイメージ空間 I と定義する。この空間の次元 ν は、データ行列のランクに一致する。また、この空

間は、 ν 次元ユークリッド空間となる。

(3) 意味射影の集合 Π_ν の設定： イメージ空間 \mathcal{I} から固有 (不変) 部分空間 (以下, “意味空間” と呼ぶ) への射影 (以下, “意味射影” と呼ぶ) の集合 Π_ν を考える. i 次元の意味空間は, $\frac{\nu(\nu-1)\cdots(\nu-i+1)}{i!}$, ($i = 1, 2, \dots, \nu$) 個存在するので, 射影の総数は, 2^ν となる. つまり, このモデルは, 2^ν 通りの文脈を表現する能力を持つ.

(4) 意味解釈オペレータ S_p の構成： 文脈を決定する ℓ 個の単語列 (以下, “文脈語群” と呼ぶ) に対応して構成されるベクトル (以下, “文脈ベクトル” と呼ぶ) s_ℓ としきい値 ε_s が与えられたとする. このとき, その文脈に応じた意味射影 $P_{\varepsilon_s}(s_\ell)$ を決めるオペレータ (以下, “意味解釈オペレータ” と呼ぶ) S_p を次のように構成する.

(a) 文脈ベクトル s_ℓ を構成する ℓ 個の単語を各々イメージ空間 \mathcal{I} へ写像する. この写像では, ℓ 個の単語を各々イメージ空間 \mathcal{I} 内でフーリエ展開し, フーリエ係数を求める. これは, 各単語と各意味素の相関を求めることに相当する.

(b) 各意味素ごとに, フーリエ係数の総和を求める. これは, 文脈ベクトル s_ℓ と各意味素との相関を求めることに相当する. また, このベクトルは, ν 個の意味素があるため, ν 次元ベクトルとなる. このベクトルを, 無限大ノルムによって正規化したベクトルを, 文脈ベクトル s_ℓ の意味重心と呼ぶ.

(c) このとき, 文脈ベクトル s_ℓ の意味重心を構成する各要素において, しきい値 ε_s を超える要素に対応する意味素を, 単語を射影する意味空間の構成に用いる. これにより, 意味射影 $P_{\varepsilon_s}(s_\ell)$ を決定する.

このオペレータは, 文脈語群と相関の高い意味空間の自動的な選択を実現する.

(5) 意味空間における距離計算： 文脈ベクトル s_ℓ により, 各意味素ごとに重みを定める. そして, 意味空間において, その重みを考慮した単語間の距離計算を行う. これにより, 文脈に応じた単語間の関係の解釈 (意味空間の選択, および, その空間内における最良近似) が可能となる.

2.2 定式化

本節では, 意味の数学モデルの定式化について述べる. 具体的データを用いた実験例は文献 (6), (7) に示している.

2.2.1 イメージ空間 \mathcal{I} の設定

ここでは, m 個の単語について各々 n 個の特徴

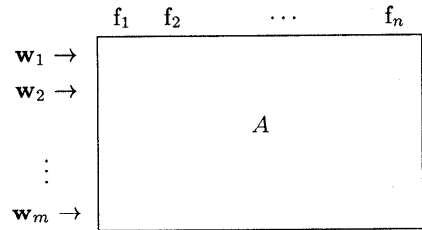


図1 データ行列 A の構成
Fig.1 Data matrix A .

(f_1, f_2, \dots, f_n) を列挙した各単語に対する特徴付きベクトル \mathbf{w}_i ($i = 1, \dots, m$) が与えられているものとし, そのベクトルを並べた m 行 n 列のデータ行列を A とする (図1).

- (1) データ行列 A の相関行列 $A^T A$ を作る.
- (2) $A^T A$ を固有値分解する.

$$A^T A = Q \begin{pmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_\nu & & \\ & & & 0 \dots 0 & \\ & & & & 0 \dots 0 \end{pmatrix} Q^T,$$

$$0 \leq \nu \leq n.$$

ここで行列 Q は,

$$Q = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)^T$$

である. この \mathbf{q}_i は, 相関行列の固有ベクトル, つまり意味素である.

- (3) このとき, イメージ空間 \mathcal{I} を以下のように定義する.

$$\mathcal{I} := \text{span}(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_\nu).$$

$(\mathbf{q}_1, \dots, \mathbf{q}_\nu)$ は \mathcal{I} の正規直交基底である.

2.2.2 意味射影集合 Π_ν の設定

P_{λ_i} を次のように定義する.

$$P_{\lambda_i} \stackrel{d}{\iff} \lambda_i \text{ に対応する固有空間への射影,} \\ \text{i.e. } P_{\lambda_i} : \mathcal{I} \rightarrow \text{span}(\mathbf{q}_i).$$

意味射影の集合 Π_ν を次のように定義する.

$$\Pi_\nu := \{0, P_{\lambda_1}, P_{\lambda_2}, \dots, P_{\lambda_\nu}, \\ P_{\lambda_1} + P_{\lambda_2}, P_{\lambda_1} + P_{\lambda_3}, \dots, P_{\lambda_{\nu-1}} + P_{\lambda_\nu}, \\ \vdots \\ P_{\lambda_1} + P_{\lambda_2} + \dots + P_{\lambda_\nu}\}.$$

Π_ν の要素の個数は 2^ν 個であり, これは 2^ν 通りの意味の様相表現ができることを示している.

2.2.3 意味解釈オペレータ S_p の構成

文脈ベクトル

$$s_\ell = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_\ell)$$

と, しきい値 ε_s ($0 \leq \varepsilon_s < 1$) が与えられたとき, 意

意味解釈オペレータ S_p は、その文脈ベクトル s_ℓ に応じて、意味射影 $P_{\varepsilon_s}(s_\ell)$ を決定する。すなわち、 $s_\ell \in T_\ell$ (ここで T_ℓ は ℓ 語によって構成される語群シーケンスのすべての集合である)、 $\Pi_\nu \ni P_{\varepsilon_s}(s_\ell)$ とすると、意味解釈オペレータ S_p は、 T_ℓ から Π_ν への作用素として定義される。また、 $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_\ell\}$ は、特徴付きベクトルであり、データ行列 A の特徴と同一の特徴を用いている。

オペレータ S_p は次のように定義される。

- (1) \mathbf{u}_i ($i = 1, 2, \dots, \ell$) をフーリエ展開する。

\mathbf{u}_i と \mathbf{q}_j の内積を u_{ij} とする。

$$u_{ij} := (\mathbf{u}_i, \mathbf{q}_j), \quad j = 1, 2, \dots, \nu.$$

ベクトル $\hat{\mathbf{u}}_i \in \mathcal{I}$ を次のように定める。

$$\hat{\mathbf{u}}_i := (u_{i1}, u_{i2}, \dots, u_{i\nu}).$$

これは、単語 \mathbf{u}_i をイメージ空間 \mathcal{I} に写像したものである。

- (2) 文脈ベクトル s_ℓ の意味重心 $\mathbf{G}^+(s_\ell)$ を求める。

$$\mathbf{G}^+(s_\ell) := \frac{\left(\sum_{i=1}^{\ell} u_{i1}, \sum_{i=1}^{\ell} u_{i2}, \dots, \sum_{i=1}^{\ell} u_{i\nu} \right)}{\left\| \left(\sum_{i=1}^{\ell} u_{i1}, \sum_{i=1}^{\ell} u_{i2}, \dots, \sum_{i=1}^{\ell} u_{i\nu} \right) \right\|_{\infty}}$$

この $\|\cdot\|_{\infty}$ は、無限大ノルムを示す。

- (3) 意味射影 $P_{\varepsilon_s}(s_\ell)$ を決定し、部分空間 (以下、意味空間と呼ぶ) を選択する。

$$P_{\varepsilon_s}(s_\ell) := \sum_{i \in \Lambda_{\varepsilon_s}} P_{\lambda_i} \in \Pi_\nu.$$

ただし $\Lambda_{\varepsilon_s} := \{i \mid (\mathbf{G}^+(s_\ell))_i > \varepsilon_s\}$ とする。

2.2.4 意味空間における距離計算

文脈語ベクトル s_ℓ が与えられたとする。また、データ x とデータ y の特徴付きベクトルを、イメージ空間に写像したベクトルを $\mathbf{x} \in \mathcal{I}$, $\mathbf{y} \in \mathcal{I}$ とする。このデータ間の距離 $\rho(\mathbf{x}, \mathbf{y}; s_\ell)$ を次のように定める。

$$\rho(\mathbf{x}, \mathbf{y}; s_\ell) = \sqrt{\sum_{j \in \Lambda_{\varepsilon_s}} \{c_j(s_\ell)(x_j - y_j)\}^2},$$

ここで、 $c_j(s_\ell)$ は、文脈ベクトル s_ℓ に依存して決まる重みであり、次のように定義する。

$$c_j(s_\ell) := \frac{\sum_{i=1}^{\ell} u_{ij}}{\left\| \left(\sum_{i=1}^{\ell} u_{i1}, \dots, \sum_{i=1}^{\ell} u_{i\nu} \right) \right\|_{\infty}}, \quad j \in \Lambda_{\varepsilon_s}.$$

このように、距離計算において、イメージ空間を構成する各意味素 (固有ベクトル) に重みづけ ($c_j(s_\ell)$) を行うことにより、 ε_s の値が小さい場合、すなわち、2.2.3 項の (3) において意味空間を構成するために選択される固有ベクトルの数が多くなる場合においても、

文脈の認識に関する ε_s の値の影響を小さくしている。

3. 意味的連想検索アルゴリズム

意味の数学モデルでは、文脈に対応して選ばれた意味空間 (部分空間) において、検索対象語 (ベクトル) 集合 W から検索キーワードに最も近い意味を持つ検索対象語を選び出すことによって意味的連想検索を行う。意味的連想検索方式では、文脈に応じた部分空間を選び、その中から意味的に近いデータを選び出すために、次に示す 2 基本機能を実現する。

3.1 基本機能 1: 部分空間選択

文脈を決定する文脈ベクトル $s_\ell = (u_1, u_2, \dots, u_\ell)$ が与えられると (u_1 から u_ℓ までの単語ベクトルは A 行列を作成したのと同じ特徴で特徴づけられている)、文脈ベクトルの意味重心を計算し、意味射影と部分空間選択を行う。

この部分空間選択は、与えられた検索キーワードの意味を確定するために行う。

Step-1: 文脈ベクトルを構成する単語ベクトルのフーリエ展開:

u_1 から u_ℓ までの各単語ベクトルについてフーリエ展開を行い、各々の意味素 (固有ベクトル) との相関を計算する。

Step-2: フーリエ展開によって得られた値を、各意味素 (固有ベクトル) ごとに合算する:

文脈語と意味素の相関を計算するために、フーリエ展開によって得られた値を各々の意味素ごとに合算することにより意味重心ベクトルを形成する。

Step-3: 部分空間の決定:

Step-2 で求めた意味重心ベクトルの各意味素上での値の絶対値が、与えられたしきい値 ε_s より大きい意味素群を選択し、部分空間を構成する。

3.2 基本機能-2: 解の検索

与えられたキーワードに意味的に近い検索対象語を、基本機能 1 によって選択された部分空間に写像された検索対象語群の中から選び出す。

Step-1: 検索語ベクトルのフーリエ展開:

検索語ベクトルのフーリエ展開を、そのベクトルを構成する要素ごとに行い、部分空間を構成する意味素に対応する値だけを、意味素ごとに加算していく。

Step-2: 検索対象語のフーリエ展開:

全検索対象語を、与えられた文脈に対応した部分空間に写像する。すなわち、全検索対象語ベクトルのフーリエ展開を行い、与えられた文脈に対応した部分空間に写像する。

Step-3: 動的な距離系による検索キーワードと検索対象語との距離計算:

2.2.4 項で定義した距離系を使用し, 検索語ベクトルと各検索対象語ベクトルとの距離を計算する. そして, 与えられたキーワードに最も距離 (意味) の近い検索対象語を, 検索対象語のベクトル群 W の中から選び出す. キーワード p が与えられたとき, 次の式を満たす検索対象語ベクトル w を選び出す.

$$\min_{w \in W} \rho(p, w; s_\ell).$$

4. 意味的連想検索の高速化アルゴリズム

3章で述べた意味的連想検索を高速化する方法としては, 距離計算の並列化による高速化や, 距離計算アルゴリズム自体の高速化が考えられる. ここでは, 本モデルにおいて, 意味の近い2つの単語は, 空間上における距離の近い2点として配置されていることを利用した高速化アルゴリズムを述べる. このアルゴリズムでは, 検索キーワードに最も近い意味の検索対象語を求めるにあたり, 検索キーワードから距離 (意味) の遠い検索対象語との距離計算を動的に省くことにより高速化を実現する.

4.1 意味的連想検索の高速化アルゴリズム

本アルゴリズムでは, あらかじめ, 検索対象語群は, 各意味素 q_i ($i = 1 \sim \nu$) ごとに, 各検索対象語の要素の大きさに応じて, ソートされていることを前提とする.

検索キーワード x の ν 次元ベクトルを

$$\mathbf{x} := (x_1, x_2, \dots, x_\nu), \quad \mathbf{x} \in I$$

とし, 検索対象語の ν 次元ベクトル \mathbf{y}_i を

$$\mathbf{y}_i := (y_{i1}, y_{i2}, \dots, y_{i\nu}), \quad \mathbf{y}_i \in I, \\ i = 1, 2, \dots, m$$

とする.

- (1) 文脈ベクトル s_ℓ に応じた部分空間 $P_{e_s}(s_\ell)I$ を形成する意味素の中から, 意味重心 $\mathbf{G}^+(s_\ell)$ に最も関連が強い意味素 q_j を選ぶ.
- (2) 範囲変数 Δ_h を ∞ に初期化する. 検索キーワードとの距離が最も近い検索対象語の候補 \mathbf{z} を $NULL$ に初期化する. 検索対象語のベクトルの集合 \mathcal{Y} を全検索対象語のベクトルの集合として初期化する.
- (3) 検索対象語のベクトルの集合 \mathcal{Y} から, $x_j \pm \Delta_h$ の範囲外にある検索対象語を除く (このステップが初めて実行される場合には, $\Delta_h = \infty$ なので検索対象語は1つも除かれない).
- (4) \mathcal{Y} が空集合ならば, 検索対象語 \mathbf{z} を解とし, 終

了する.

- (5) 集合 \mathcal{Y} の要素から, 意味素 q_j 上における, 検索キーワードのフーリエ係数 x_j に最も近いフーリエ係数 y_{kj} ($1 \leq k \leq m$) を持つ検索対象語 $\mathbf{y}_k \in \mathcal{Y}$ を求める (この検索は, 検索対象語を意味素 q_j 上における値の大きさの順にソートしていることにより, 高速に実行できる). \mathbf{y}_k を集合 \mathcal{Y} から取り除く.
- (6) 文脈に応じた部分空間上で, 検索キーワードのベクトル \mathbf{x} と検索対象語のベクトル \mathbf{y}_k との距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ を求める.

もし, 範囲変数 Δ_h より距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ の方が大きいならば, (4) に行く.

さもなければ, つまり, 範囲変数 Δ_h よりも距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ の方が, 小さいならば, 範囲変数 Δ_h を $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ とし, ベクトル \mathbf{y}_k を検索キーワードとの距離が最も近い検索対象語の候補 \mathbf{z} とする. そして, (3) に行く.

上述の (3) の処理によって, 連想検索時における検索対象語の数を大幅に減らすことが可能となる.

4.2 上記アルゴリズムによる意味的連想検索動作の例

ここでは, 上記アルゴリズムを適用した高速意味的連想検索の動作 (図2) を示す.

- (a) 文脈ベクトル s_ℓ が与えられ, 文脈と関連の強い q_i と q_j の2軸が選ばれたとする. そして, それらの軸から構成される2次元の部分空間を文脈に対応する意味空間として設定し, 検索キーワード x と検索対象語 $\mathbf{y}_1, \dots, \mathbf{y}_8$ を射影する.
- (b) 選択した意味空間を構成する軸の中で最も重みの大きい軸 q_j 上において, 検索キーワード x の軸 q_j 上での値 x_j に最も近い値 (y_{5j}) を持つ検索対象語 (\mathbf{y}_5) を探す. 部分空間上での距離 $\rho(\mathbf{x}, \mathbf{y}_5; s_\ell)$ を計算し, 軸 q_j 上において, 距離 $\rho(\mathbf{x}, \mathbf{y}_5; s_\ell)$ よりも離れている検索対象語群 $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_8\}$ を検索対象から取り除く. 検索キーワードに最も近い単語の候補 (\mathbf{y}_5) を選ぶ. \mathbf{y}_5 を検索対象語群から除外する.
- (c) 軸 q_j 上において, 検索キーワード x の軸 q_j 上での値 x_j に次に近い値 (y_{6j}) を持つ検索対象語 (\mathbf{y}_6) を探し, 部分空間での距離 $\rho(\mathbf{x}, \mathbf{y}_6; s_\ell)$ を計算する. $\rho(\mathbf{x}, \mathbf{y}_5; s_\ell)$ と $\rho(\mathbf{x}, \mathbf{y}_6; s_\ell)$ を比較し, ここでは, $\rho(\mathbf{x}, \mathbf{y}_6; s_\ell)$ の方が距離が小さいので, 検索キーワードに最も近い単語の候補を \mathbf{y}_6 に更新する. \mathbf{y}_6 を検索対象語群から除外する. 前に取り除いた検索対象語群 $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_8\}$ に加えて, 軸 q_j 上において, 距離 $\rho(\mathbf{x}, \mathbf{y}_6; s_\ell)$ よりも離れている検

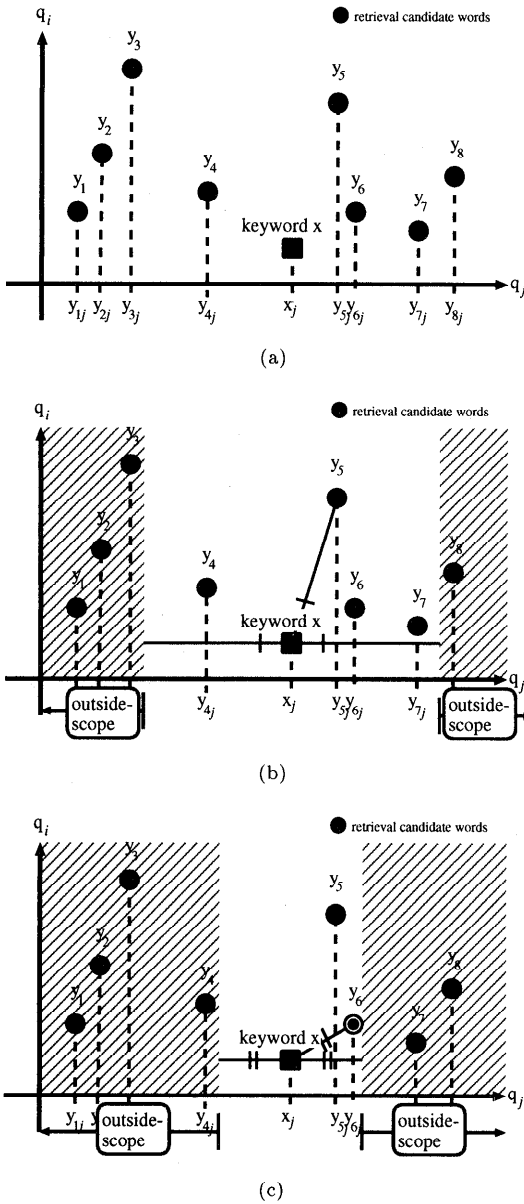


図2 検索キーワードに最も近い検索対象語を高速に求めるアルゴリズムの動作例

Fig.2 An example of execution by the algorithm for extracting the closest data item to the keyword.

検索対象語群 $\{y_4, y_7\}$ を検索対象から取り除く。ここで、検索対象語群がなくなるので、 y_6 を解とする。

5. 意味的連想検索の高速化アルゴリズムの拡張

パターン・マッチングによる連想検索では、利用者が与えたパターンと同じ文字列に対応する情報を求め

る。一方、意味の数学モデルによる意味的連想検索では、利用者が与えた検索キーワードに意味的に近い情報を検索することが可能となる。このような意味的連想検索においては、意味的に最も近い単語だけでなく、2番目、3番目に近い単語も、利用者にとっては有用である。

検索キーワードに意味的に最も近い単語を求めるアルゴリズムは、解を求めるのに必要のない距離計算を省くことにより、高速化を実現している。

しかし、最も近い単語を求める過程で、本来2番目に近い単語の距離計算を省く場合があり得るので、検索キーワードからの距離が最も近い単語の次の候補であった単語が、検索キーワードから2番目に近いとは限らない。

検索キーワードに2番目に近い単語を求める方法としては、検索キーワードとの距離が最も小さい検索対象語を検索対象から除き、先のアルゴリズムを繰り返す方法が考えられる。しかし、この方法では、同一の検索対象語に対して、同一の距離計算を再度行ってしまう可能性がある。

次のアルゴリズムは、検索キーワードに最も近い検索対象語を求めるアルゴリズムを拡張し、任意の個数の解を高速に求めるものである。

5.1 意味的連想検索の高速化アルゴリズムの拡張

本アルゴリズムでは、あらかじめ、検索対象語群は、各意味素 q_i ($i = 1 \sim \nu$) ごとに、各検索対象語の要素の大きさに応じて、ソートされていることを前提とする。

検索キーワード x の ν 次元ベクトルを

$$\mathbf{x} := (x_1, x_2, \dots, x_\nu), \quad \mathbf{x} \in \mathcal{I}$$

とし、検索対象語の ν 次元ベクトル \mathbf{y}_i を

$$\mathbf{y}_i := (y_{i1}, y_{i2}, \dots, y_{i\nu}), \quad \mathbf{y}_i \in \mathcal{I}, \\ i = 1, 2, \dots, m$$

とする。

- (1) 文脈ベクトル s_ℓ に応じた部分空間 $P_{e_s}(s_\ell)\mathcal{I}$ を形成する意味素の中から、意味重心 $\mathbf{G}^+(s_\ell)$ に最も関連が強い意味素 q_j を選ぶ。
- (2) 検索キーワードからの距離の小さい順に並べられた解の候補リスト \mathcal{Z} を $NULL$ に初期化する。検索対象語のベクトルの集合 \mathcal{Y} を、全検索対象語のベクトルの集合として初期化する。
- (3) 解の候補リスト \mathcal{Z} の先頭要素を、解の候補 \mathbf{z} とする。
- (4) 解の候補 \mathbf{z} が $NULL$ ならば、範囲変数 Δ_h を ∞ に初期化する。解の候補 \mathbf{z} が $NULL$ でなければ、範囲変数 Δ_h を、検索キーワードと解の候補 \mathbf{z}

との部分空間 $P_{e_s}(s_\ell)I$ 上での距離 $\rho(\mathbf{x}, \mathbf{y}_z; s_\ell)$ とする。

- (5) 集合 \mathcal{Y} の要素から, $x_j \pm \Delta_h$ の範囲内にあり, かつ, 意味素 \mathbf{q}_j 上において, 検索キーワードのフーリエ係数 x_j に最も近いフーリエ係数 y_{kj} ($1 \leq k \leq m$) を持つ検索対象語のベクトル $\mathbf{y}_k \in \mathcal{Y}$ を求める (この検索は, 検索対象語を意味素 \mathbf{q}_j 上における値の大きさの順にソートしていることにより, 高速に実行できる). もしも \mathbf{y}_k が存在するならば, 集合 \mathcal{Z} から取り除く. もし, \mathbf{y}_k が存在しなければ, 解の候補リスト \mathcal{Z} から \mathbf{z} を取り除き, \mathcal{Z} を解とする. 解の個数が指定個数に至っていないならば, (3) に行く. 指定個数の解が得られたならば, 終了する.

- (6) 文脈に応じた部分空間上で, 検索キーワードのベクトル \mathbf{x} と検索対象語のベクトル \mathbf{y}_k との距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ を求める. 解の候補リスト \mathcal{Z} に, \mathbf{y}_k を追加する.

距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ が, 範囲変数 Δ_h より大きければ, (5) に行く. 距離 $\rho(\mathbf{x}, \mathbf{y}_k; s_\ell)$ が, 範囲変数 Δ_h より小さければ, ベクトル \mathbf{y}_k を解の候補 \mathbf{z} とする. そして, (4) に行く.

5.2 拡張されたアルゴリズムによる意味的連想検索動作の例

上記アルゴリズムを適用した高速意味的連想検索の動作例を示す. ここでは, 高速な連想検索のアルゴリズムによって, 検索キーワードから最も近い単語の検索を求めた直後の状態を考える.

図3は, $\mathbf{y}_5, \mathbf{y}_6$ の順に距離計算を行い, 検索キーワードのベクトル \mathbf{x} に最も近い検索対象語 \mathbf{y}_6 を求めた直後の状態である.

- (a) 検索キーワードのベクトル \mathbf{x} に2番目に近い単語を求めるにあたって, 先ほど求めた, 検索キーワードに最も近い解である \mathbf{y}_6 を検索対象語群から取り除く.

現在の時点で距離が分かっている検索対象語のベクトル群 $\{\mathbf{y}_3\}$ の中で, 検索キーワードのベクトル \mathbf{x} からの距離が最も小さいベクトルは \mathbf{y}_5 であることが分かる. これにより, 斜線内の領域には, 検索キーワードのベクトル \mathbf{x} からの距離が \mathbf{y}_5 より小さい検索対象語は存在しないことが分かる. この領域内の検索対象語群 $\{\mathbf{y}_5\}$ を検索対象から取り除く.

- (b) 軸 \mathbf{q}_j 上において, 距離 $\rho(\mathbf{x}, \mathbf{y}_5; s_\ell)$ よりも離れている検索対象語群 $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_8\}$ を検索対象から取り除く. 検索キーワードに最も近い単語の候補を \mathbf{y}_5 とする.

- (c) 意味空間を構成する軸の中で最も重みの大きい

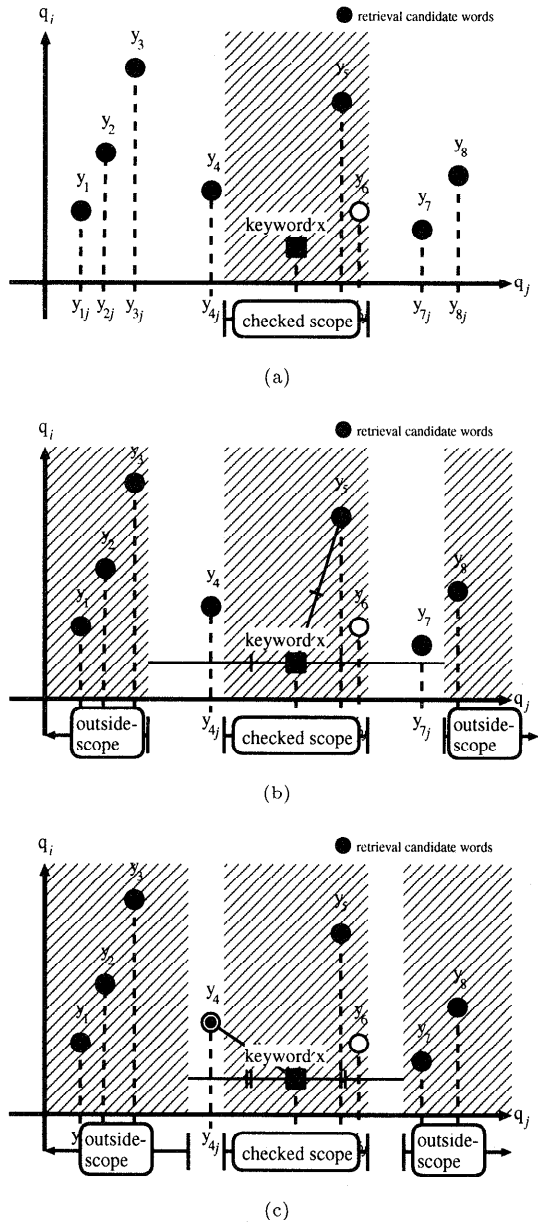


図3 検索キーワードに2番目に近い単語を探すアルゴリズムの動作例

Fig. 3 An example of execution by the algorithm for extracting the secondary closest data item to the keyword.

軸 \mathbf{q}_j 上において, 検索キーワード x の軸 \mathbf{q}_j 上での値 (x_j) に最も近い値 (y_{4j}) を持つ検索対象語 (\mathbf{y}_4) を求める. 意味空間での距離 $\rho(\mathbf{x}, \mathbf{y}_4; s_\ell)$ を計算する. $\rho(\mathbf{x}, \mathbf{y}_5; s_\ell)$ と $\rho(\mathbf{x}, \mathbf{y}_4; s_\ell)$ を比較し, $\rho(\mathbf{x}, \mathbf{y}_4; s_\ell)$ の方が距離が小さいので, 検索キーワードに最も近い単語の候補を \mathbf{y}_4 に更新する. \mathbf{y}_4 を

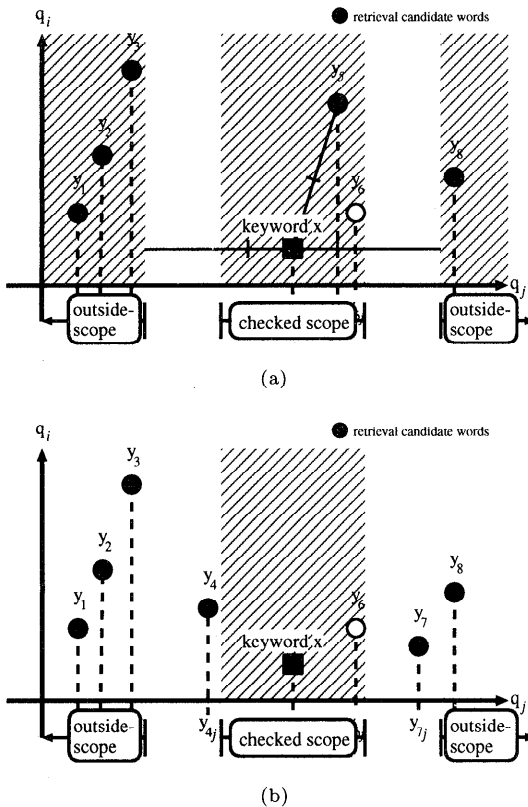


図4 検索キーワードに2番目に近い単語を探すアルゴリズムの動作の特殊な場合

Fig.4 An exceptional example of the algorithm for extracting the secondary closest data item to the keyword.

検索対象語群から取り除く。前に取り除いた検索対象語群 $\{y_1, y_2, y_3, y_8\}$ に加えて、軸 q_j 上において、距離 $\rho(x, y_4; s_\ell)$ よりも離れている検索対象語群 $\{y_7\}$ を検索対象から取り除く。ここで、検索対象語群がなくなるので、 y_4 を解とする。

本アルゴリズムの動作における特殊な例を図4に示す。図4(a)は前回の距離計算で求めた距離を使用して、計算の必要のない検索対象語群を排除した時点で、距離計算を行うべき検索対象語が存在しなくなった場合の例である。この場合には、即座に解が y_5 に決まる。図4(b)は前回の距離計算で求めた中で、2番目に検索キーワードになる候補となる検索対象語が存在しない場合である。この場合は、最も重みの大きい軸上でキーワード x の軸上で値 x_j に最も近い値 y_{4j} を持つ検索対象語 y_4 を見つけ出し、それを最初の候補とする。

本アルゴリズムの適用を繰り返すことにより、検索キーワードからの距離が近い順に、任意の個数の解を

求めることが可能となる。

6. 基本英単語を対象とした実験

意味の数学モデルによる連想検索では、検索キーワードと文脈の組合せにより、無限に近いパターンの検索を行うことが可能であるので、意味的連想検索の高速アルゴリズムの有効性を検証するにあたって、すべての検索パターンを試行するのは不可能である。そこで、Longman Dictionary of Contemporary English¹¹⁾における基本英単語2328語を対象として、それぞれの単語が1回ずつ解となることを想定した実験を行った。

6.1 実験環境

実験では、限られた数の基本英単語のみを使用して単語の定義を行っている英英辞書を参照した。具体的には、Longman Dictionary of Contemporary Englishにおける基本英単語2328語を、The General Basic English Dictionary¹⁴⁾の定義を用いて定義し、単語の活用形を基本形に戻すフィルタを通して 2328×874 の行列を作成し、イメージ空間を構成した。検索対象単語群として、上述の基本英単語2328語を用いた。上の行列において、同じ見出し語を持つ単語群の定義を合成した行列を使用して、検索キーワードとなる単語群の定義を行った。

実験に使用した計算機はSun4/ELC。OSはSunOS 4.1.4である。

6.2 実験方法

実験を次の方法によって行った。

- (1) 検索対象語群から、1単語を選び出し、それを d_i とする。
- (2) d_i の定義に使用されている語句を、文脈語群とする。
- (3) d_i と同じ綴の単語を、検索キーワードとする。
- (4) 検索対象語群の中から、与えた文脈において、検索キーワードに近い順に、10単語検索する。

図5は、実験の一部を示したものである。この例において、「ring」には「指輪」の意味の「ring1」と、「鳴る」の意味の「ring2」が存在する。この図では、「ring1」を解と想定している。このとき、検索キーワードには、「指輪」の意味の「ring1」と、「鳴る」の意味の「ring2」のベクトルを合成したベクトルを与える。また、ここでは、文脈語群として、「ring1」の定義に使用されている単語群を与える。検索対象語2328単語の中から、検索キーワードに近い順に10単語を検索し、そのときに行った距離計算回数によってアルゴリズムの効果を評価する。

見出し語	説明語
right	(of act, feeling etc.) in agreement with what has approval of so ciety, one's better self
<u>ring1</u>	<u>band of material forming circle, one of gold etc. for finger</u>
ring2	give out clear sound (as) of blow on metal
risk	danger, chance of bad outcome
river	body of moving water in naturally formed bed
⋮	⋮
解と想定した単語：	<u>ring1</u>
検索キーワード：	ring (= ring1 + ring2)
文脈語群：	<u>band of material forming circle, one of gold etc. for finger</u>

図5 基本英単語を対象とした実験におけるキーワードと文脈語の生成方法

Fig.5 Creation of a keyword and context words in the experiment for basic English words.

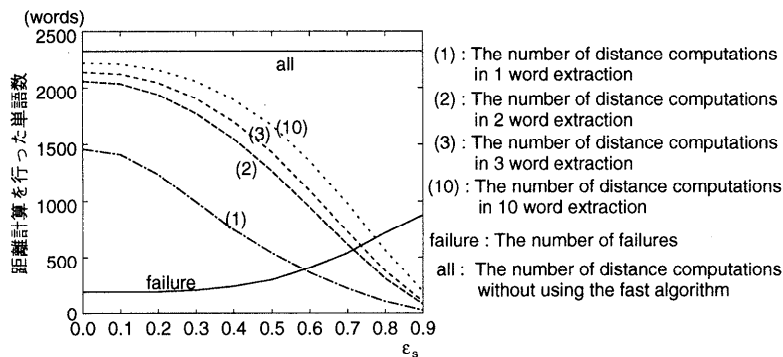


図6 基本英単語に関する実験結果

Fig.6 The experimental result for basic English words.

上記基本英単語 2328 語の各語を解とする実験を行い、それに要した平均距離計算回数を求めた。また、意味空間のしきい値 ϵ_s を、0.0 から 0.9 まで 0.1 きざみで変化させ、それによる影響を調べた。

6.3 実験結果

実験結果を図6に示す。各グラフは、1回の検索において抽出するデータ数に対応しており、(1)は第1位のデータだけを抽出する場合、(2)は第1位および第2位の、(10)は第1位から第10位のデータを抽出する場合を示している。また、“failure”に対応するグラフは、解に想定した検索対象語以外のデータが抽出された数を示している。“all”は高速化のアルゴリズムを使用しなかったときの計算回数であり、キーワードと全検索対象語との距離を計算したときの計算回数に等しい。横軸は ϵ_s の設定に対応している。

検索キーワードに最も近い検索対象語だけを求めた場合と比較すると、2番目、3番目に近い検索対象語、すなわち、2個、3個の解を求めたときには、より多

くの距離計算が必要であったことが分かる。しかし、その場合においても、すべての検索対象語との距離計算は必要ないことが分かる。

また、全体的な傾向として、意味空間のしきい値 ϵ_s の増加にともない、距離計算回数が減少している。これは、意味空間のしきい値 ϵ_s の増加にともない、距離計算を行う空間の次元数が減少していることにより、本アルゴリズムの有効性が、より強く現れることによる。しかし、 ϵ_s の増加にともない、想定した解と異なる解が選択される回数 (“failure”に対応するグラフ)も増加を始めるので、 ϵ_s に関しては、0.5以下に設定するのが望ましいと考えられる。 ϵ_s が 0.5 のとき、第1位のデータを抽出するのに、全検索対象語の約 1/4 の距離計算回数で解が求められている。これは、提案した高速化のアルゴリズムにより 4 倍の高速化を実現できたことを意味している。また、第10位までのデータを抽出するには、ほぼ 3/4 の距離計算回数で解が求められている。これは、提案した高速化アルゴ

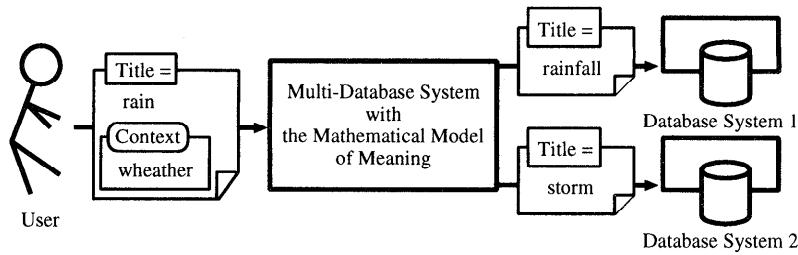


図7 意味の数学モデルのマルチデータベースへの適用
Fig. 7 Semantic Associative search for multi databases.

リズムにより 1.3 倍の高速化が実現できたことを意味している。ただし、“failure”に関しては、意味の数学モデルに学習機構を組み込むことにより、改善することが考えられる⁸⁾。

本アルゴリズムにより、求める解の個数に応じて距離計算回数は多くなるが、本アルゴリズムを用いない場合との比較において、本アルゴリズムにより距離計算回数を軽減できることが明らかとなった。

7. マルチデータベース・システムへの適用実験

本意味的連想検索方式を、複数のデータベース・システムの支援システムに適用することにより、マルチデータベース・システム^{1),17)}の検索系の主要な機能を実現することが可能になる。本意味的連想検索方式を適用したマルチデータベース・システムでは、利用者の問合せ中のキーワードを、各々のデータベース中に存在する、意味の近いキーワードに変換することができる。通常のデータベース・システムでは、利用者が発行した検索キーワードと同じ表記のデータが存在しなければ、問合せは失敗してしまう。本方式を適用し、利用者の問合せ中のキーワードを、データベース中に存在する意味の近いキーワードに変換することにより、表記を意識することなく、意味の近いデータを引き出すことが可能になる。つまり、利用者は、自分の求めるデータが、各々のデータベース中で、どのような表現で存在しているかを意識せずに、意味的連想検索を行うことが可能になる(図7)。この方式により、問合せ中のキーワードの変換を自動的に行うことができるので、マルチデータベース・システムが、異なるデータベース群の各々に問合せを発行することにより、情報を引き出すことが可能になる。

本実験では、対象データベース・システムとして、筑波大学内にある UTOPIA データベース群¹⁹⁾を利用した。UTOPIA データベース・システムは、様々なデータベースを含んだシステムであり、マルチデータベースの環境を実験する対象として適している。対象

となっているデータベース群は、主に図書についてのものである。

実験では、UTOPIA 内から利用できる ENV¹⁹⁾と POLL¹⁹⁾(以下 POL とする)という2つのデータベースを対象とした。ENV は環境破壊についてのデータベースであり、POL は環境汚染についてのデータベースである。

マルチデータベース・システムへの適用時には、問合せ中の属性名に関する変換への適用と、問合せ中の属性値に関する変換への適用が考えられるが、本実験では、後者への適用を行った。

7.1 実験環境

本実験では、Longman Dictionary of Contemporary English¹¹⁾において基本語とされている英単語 2328 種類を、The General Basic English Dictionary¹⁴⁾を用いて定義し、2328 × 874 の行列を作成し、イメージ空間を構成した。また、 ϵ_s は 0.2 に設定した。

実験に使用した計算機は Sun4/ELC。OS は SunOS 4.1.4 である。

7.2 実験方法

ここでは、ENV と POL のデータベース中で、本のタイトル中に“TSUKUBA”または“KASUMI-GAURA”を含む、という条件のもとで行った。その結果、ENV データベースからは 45 の属性値(表1)、POL データベースからは 49 の属性値(表2)が存在し、これらの属性値を検索対象データ項目群とした。

検索対象データ項目については、The General Basic English Dictionary の説明語を用いて定義し、ベクトルを形成した。また、利用者が使用可能なキーワードについては、Longman Dictionary of Contemporary English における基本英単語 2328 単語の各々について、The General Basic English Dictionary の説明語を用いて定義し、ベクトルを形成した。そして、同名異義を持つ見出し語については、その各説明語を合成して、登録した。これにより、使用可能キーワードは

表1 ENV (ENVIRONMENT) データベースにおける検索対象データ項目

Table 1 Retrieval candidate data items used in the ENV database.

academic	area	basin	bedrock	bloom	building
case	chemical	chemistry	composition	concentrations	concept
construction	crystalline	discharge	dissolved	distributions	drainage
dynamic	dynamics	effect	element	events	influence
lake	model	new	oxygen	particles	quality
rainfall	resources	seasonal	selected	size	streams
studies	succession	temporal	town	trace	tunnel
urban	variations	water			

表2 POL (POLLUTION) データベースにおける検索対象データ項目

Table 2 Retrieval candidate data items used in the POL database.

accessory	alga	application	aquatic	atmospheric	bay
behaviour	bloom	bog	carbon	case	chemical
city	community	composition	cost	damage	debris
different	distributions	dynamic	effect	estimating	impact
indicated	lake	load	losses	model	models
musty	nutrient	organic	particles	plant	population
rapid	ratio	role	seasonal	secondary	size
species	storm	succession	treatment	uptake	variations
welfare					

表3 マルチデータベースに関する実験結果

Table 3 Experimental results for multi databases.

実験結果1 (keyword: rain, context: weather)

ENV database				POL database			
順位	Selected Data Item	Distance	R	順位	Selected Data Item	Distance	R
1	rainfall	0.521618	3/45	1	storm	0.494476	1/49
2	streams	0.566394	0/45	2	atmospheric	0.695877	3/49
3	oxygen	0.595867	0/45	3	plant	0.761912	2/49

R = 距離計算回数 / 全検索対象データ項目数

実験結果2 (keyword: water, context: weather)

ENV database				POL database			
順位	Selected Data Item	Distance	R	順位	Selected Data Item	Distance	R
1	water	0.733407	1/45	1	storm	0.974418	1/49
2	oxygen	1.218185	3/45	2	atmospheric	1.443099	3/49
3	rainfall	1.277171	0/45	3	plant	1.512229	1/49

R = 距離計算回数 / 全検索対象データ項目数

実験結果3 (keyword: water, context: pool)

ENV database				POL database			
順位	Selected Data Item	Distance	R	順位	Selected Data Item	Distance	R
1	water	1.438020	45/45	1	lake	1.971830	49/49
2	streams	1.887675	0/45	2	aquatic	2.112745	0/49
3	lake	1.971830	0/45	3	alga	2.136467	0/49

R = 距離計算回数 / 全検索対象データ項目数

2115単語となった。

実験では、利用者の問合せ中のキーワードを、各々のデータベース中に存在するキーワードに変換するときにおける、高速な意味的連想検索アルゴリズムの有効性を調べた。

7.3 実験結果

実験結果を表3に示す。実験結果1は、利用者が検索キーワードとして“rain”を与え、そのときの文脈として“weather”を与えたときの結果である。ENVデータベース中には“rain”という表記のデータは存在していないので、通常のデータベース検索では、この

問合せは失敗してしまう。しかし、意味の数学モデルによって、ENVデータベース中に存在するキーワードのうちで、“rain”に最も意味の近いデータ項目（検索対象データ項目として登録されている）は“rainfall”であるという結果が抽出されている。そこで、“rain”というキーワードを“rainfall”というキーワードに変換して問合せを発行することにより、関連する情報を引き出すことが可能となる。POLデータベースにおいても、同様のことを行うことができる。POLデータベース中にも“rain”という表記のデータは存在していない。しかし、意味の数学モデルによって、POLデータベース中に存在するデータ項目の中で、“rain”に最も意味の近いデータ項目は“storm”であるという結果が抽出されている。そこで、“rain”というキーワードを“storm”に変換して問合せを発行することにより、その関連情報を引き出すことができる。

本実験システムにおいて、意味的連想検索に要した距離計算回数に注目する。Rの分母は全検索対象データ数を表しており、提案アルゴリズムを適用しなかったときの距離計算回数を表している。また、Rの分子は高速化アルゴリズムを適用した場合に解を求めるのに要した距離計算回数を表している。ENVデータベースでは45個の検索対象のデータ項目について、わずか3回の距離計算で、3つの解を求めることが可能であったことを示している。POLデータベースにおいても、49個の検索対象のデータ項目について、6回の距離計算で、3つの解を求めることが可能であったことを示している。それぞれ、15倍、8倍の高速化が実現できていることを示している。

実験結果2と3は、意味的な関係が文脈に応じて変化する例である。実験結果2、実験結果3とも、キーワードは“water”であるが、POLデータベースでは、文脈が“weather”のときには“storm”が選択され、文脈が“pool”のときには“lake”が選択されている。ENVデータベースには、“water”というデータが存在するので、それが選択されている。もし、ENVデータベースを対象として、“water”以外のキーワードを使用する場合には、3番目の検索結果である“rainfall”や“lake”を採用する。

実験結果2において、3つの解を求めるのに必要な距離計算の回数は、ENVデータベースで4回、POLデータベースで5回の距離計算が必要であった。これらは、それぞれ、11倍、10倍の高速化が実現できていることを示している。

実験結果3は、意味的連想検索の高速化アルゴリズムが有効に働かなかった場合、すなわち、ENVデータ

ベースで45個の検索対象データ項目に対して45回、POLデータベースで49個の検索対象データ項目に対して49回の距離計算が行われた場合である。この場合においても、距離計算回数が、本提案アルゴリズムを使用しない場合を超えることはない。

8. 結 論

本論文では、文脈や状況に応じて、検索キーワードに最も近い意味の単語の連想検索を可能とする意味の数学モデルにおいて、意味的に近い任意の個数の単語を高速に求めるアルゴリズムを提案した。また、上のアルゴリズムを適用した高速な意味的連想検索システムにより、基本英単語を対象とした実験を行い、提案アルゴリズムの有効性を確認した。さらに、実在するデータベース群を対象としたマルチデータベース・システムへの適用実験を行い、高速な意味的連想検索のアルゴリズムが有効であることを確認した。

今後は、本論文で提案した高速アルゴリズム、および、意味的連想検索のための学習機構の実現を行っていく予定である。

参 考 文 献

- 1) Bright, M.W., Hurson, A.R. and Pakzad, S.H.: A Taxonomy and Current Issues in Multidatabase System, *IEEE Computer*, Vol.25, No.3, pp.50-59 (1992).
- 2) David, R. and Lenat, D.B.: *Knowledge-based Systems in Artificial Intelligence*, McGraw-Hill (1982).
- 3) Gallant, S.I.: A Practical Approach for Presenting Context and for Performing Word Sense Disambiguation Using Neural Networks, *Neural Computation*, Vol.3, pp.293-309 (1991).
- 4) Guttman, A.: R-trees: A Dynamic Index Structure for Spatial Searching, *Proc. ACM SIGMOD*, pp.47-57 (June 1984).
- 5) 伊理正夫 (監修), 腰塚武志 (編): 計算幾何学と地理情報処理, 第2版, 共立出版 (1993).
- 6) Kitagawa, T. and Kiyoki, Y.: The Mathematical Model of Meaning and Its Application to Multidatabase Systems, *Proc. 3rd IEEE International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, pp.130-135 (April 1993).
- 7) Kiyoki, Y., Kitagawa, T. and Hitomi, Y.: A Fundamental Framework for Realizing Semantic Interoperability in a Multidatabase Environment, *Journal of Integrated Computer-Aided Engineering*, Vol.2, No.1, pp.3-20, John Wiley & Sons (1995).

- 8) 清木 康, 北川高嗣, 宮原隆行, 倉田佳世子: 意味の数学モデルによる意味的連想処理方式と学習機構, 信学技報 DE95-8, pp.57-64 (1995).
- 9) Kolodner, J.L.: *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum Associates (1984).
- 10) Krikelis, A. and Weems, C.C.: Associative Processing and Processors, *IEEE Computer*, Vol.27, No.11, pp.12-17 (1994).
- 11) *Longman Dictionary of Contemporary English*, Longman (1987).
- 12) Natural Language Processing, *Comm. ACM*, Vol.39, No.1 (1996).
- 13) Nick, R., Stephen, K. and Frederic, V.: Nearest Neighbor Queries, *Proc. ACM SIGMOD*, pp.71-75 (1995).
- 14) Ogden, C.K.: *The General Basic English Dictionary*, Evans Brothers Limited (1940).
- 15) Potter, J.L.: *Associative Computing*, Frontiers of Computer Science Series, Plenum (1992).
- 16) Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W. and Harshman, R.A.: Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, Vol.41, No.6, pp.391-407 (1990).
- 17) Sheth, A. and Larson, J.A.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, *ACM Computing Surveys*, Vol.22, No.3, pp.183-236 (1990).
- 18) 上坂吉則, 尾関和彦: パターン認識と学習のアルゴリズム, 文一総合出版 (1990).
- 19) UTOPIA 利用の手引, 筑波大学学術情報処理センター (July 1992).

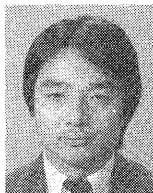
(平成 8 年 8 月 28 日受付)

(平成 9 年 5 月 8 日採録)



宮原 隆行 (学生会員)

1970 年生. 1994 年筑波大学第三学群情報学類卒業. 現在同大学院博士課程工学研究科電子・情報工学専攻に在籍中. データベースシステムの研究に従事.



清木 康 (正会員)

1978 年慶應義塾大学工学部電気工学科卒業. 1983 年同大学院工学研究科博士課程修了. 工学博士. 同年, 日本電信電話公社武蔵野電気通信研究所入所. 1984~1995 年筑波大学電子・情報工学系講師, 助教授を経て, 1996 年より慶應義塾大学環境情報学部助教授, 現在に至る. データベースシステム, 知識ベースシステム, 分散・並列処理システムの研究に従事. ACM, IEEE, 電子情報通信学会, 日本ソフトウェア科学会各会員.



北川 高嗣

1978 年名古屋大学工学部卒業. 1983 年同大学院工学研究科博士課程修了. 工学博士. スタンフォード大学計算機科学科客員研究員, 愛媛大学理学部数学科講師を経て 1990 年より筑波大学電子・情報工学系に勤務. 現在同学系助教授. 数値解析, 逆問題, マルチメディア情報システムの研究に従事. 日本応用数理学会会員.