# 汎用連想計算エンジンの開発と大規模文書分析への応用

## Development of the generic association engine for processing large corpora

高野　明彦 [1)]　西岡　真吾 [2)]　今一　修 [2)]　岩山　真 [2)]
Akihiko TAKANO　Shingo NISHIOKA　Osamu IMAICHI　Makoto IWAYAMA
aki@nii.jp　{nis,imaichi,iwayama}@harl.hitachi.co.jp

丹羽　芳樹 [2)]　久光　徹 [2)]　藤尾　正和 [2)]
Yoshiki NIWA　Toru HISAMITSU　Masakazu FUJIO
{yniwa,hisamitu,m-fujio}@harl.hitachi.co.jp

徳永　健伸 [3)]　奥村　学 [4)]　望月　源 [5)]　野本　忠司 [6)]
Takenobu TOKUNAGA　Manabu OKUMURA　Hajime MOCHIZUKI　Tadashi NOMOTO

1) 国立情報学研究所 ソフトウェア研究系 プログラミング言語研究部門
2) (株) 日立製作所、中央研究所（〒350-0395 埼玉県比企郡鳩山町）
3) 東京工業大学 情報理工学研究科 計算工学専攻
4) 東京工業大学 精密工学研究所 知能化工学部門
5) 北陸先端科学技術大学院大学 情報科学研究科 自然言語処理学
6) 国文学研究資料館 研究情報部 情報メディア室

**ABSTRACT.** Generic Engine for Transposable Association (GETA) has been developed aiming at an efficient tool for manipulating very large dimensional sparse matrices, which typically appear as index files for large scale text retrieval. This engine can be directly used to realize associative searching systems, which accept a group of texts as queries, and return highly related texts in the relevance order. The usefulness of this type of associative search has long been recognized because that covers weakness of ordinary key-word searching which sometimes returns no hit and sometimes too many hits. But a serious problem of associative search is its very high computational cost, and that has long prevented associative search from prevailing as a standard searching method.

GETA is expected to solve this problem. We built an experimental associative searching system using GETA for about one million texts, and verified that real-time associative search (less than 10 sec. response time) is possible with an ordinary single CPU PC. In order for higher scalability, we have also developed parallel processing type of GETA, with which a real time associative search is possible up to about 10 million texts, using, for example, 8-nodes PC cluster.

The use of GETA is not limited to associative search, but it can be applied to a large variety of text processing techniques, such as text categorization, text clustering, and text summarization. We hope this tool will be used for accelerating research and practical application of these and other related text processing techniques.

## 1　はじめに

新聞や百科事典など文書数が数 10 万件を超える文書集合から、必要な情報をすばやく検索できる技術が強く求められている。現在実用となっているキーワード検索は、低い再現率（求めている情報が検索されない）と低い適合率（求めていない情報が大量に検索される）という問題を抱えている。これらを解決するため、我々はこれまで、検索時にキーワード集合ではなく文書それ自身を入力し、入力文書 (群) と類似の文書を検索する文書連想検索 [43] を提案し、そのための文書クラスタリング手法 [30, 31] を開発してきた。また、検索結果の内容的把握を助けるための文書や文書群の要約手法 [35, 37] を提案した。さらに、文書連想検索と自動要約機能の有機的連携を可能にする文書検索システム [34, 35, 42, 46] を自ら開発実装することを通じて、次世代情報検索技術の実用化の可能性を追求してきた。その結果、連想計算（文書群同士、単語群同士、文書群と単語群間の類似性関連性計算）こそが最も基本的で重要な計算機構であると確信するに至った。これらの研究的蓄積を踏まえ、本研究開発では、大規模な文書集合について各種の連想計算を高精度かつ高速に処理できる汎用連想計算エンジンを開発した。また、作成した汎用連想計算エンジンを研究上のツールとして用い、動的クラスタリング・要約手法、計量的語彙モデルに基づく語彙分析手法について研究を推進しつつある。さらに、それらの研究成果を利用して、汎用連想計算エンジンの改良を行った。

**汎用連想計算エンジンの開発**

文書クラスタリング、文書連想検索、特徴語抽出の計算で必要だった文書群同士・単語群同士の類似性計算を拡張して、各種の統計的計量に基づく連想計算モデルを作成している。文書群を単語のマルチセットで、単語群を文書のマルチセットでモデル化する場合のように、数学的に双対

(Dual) の関係にある２つの統計的計量が同時に必要とされることが多い。この一組の双対な計量についての連想計算が単一のインデックスを用いて同等に高速処理できるよう (双対な) データ構造を設計した。文書クラスタリングでは、各文書は文書数が１件の文書群 (クラスタ) と見なされるので、文書と文書群を同等に扱えるデータ構造設計が必須となる。このような性質を備えた基本データ構造を高度に圧縮して保持する高速な汎用連想計算エンジンを作成した。

**動的クラスタリング・要約手法の開発**

上記の汎用連想計算エンジンを用いて、精度の犠牲なしに文書クラスタリングや文書連想検索を高速に処理する手法を検討した。高速文書クラスタリングの実現により、検索結果の自動分類や、検索結果の要約を検索者に提示するなど、高度な検索インターフェイスが可能になる。本研究項目では，オンラインで高速に動作する高速文書クラスタリングのプロトタイプを汎用連想計算エンジンを用いて実装した。また、検索結果の動的な分類、および要約生成の可能性についても検討した。要約生成にあたっては、語の話題性に関する語彙分析手法の成果の利用を検討する。検索結果のみならず文書群の適切な要約は、入力文書数が極端に多い文書連想検索にとって必須の技術と考えられる。(e.g. 部分 DB 間の連想)

**計量的語彙モデル、語彙分析手法の開発**

情報検索において単語の果たす役割は想像以上に大きい。人間は語という短い文字列から、自らの知識や経験に基づく多くのものを意識的無意識的に想起できる。我々はこの観点から、検索結果の文書群を特徴づける単語集合の自動抽出方法 [34, 35, 36, 46] を提案してきた。また、「語はその出現環境により特徴づけられる」という考え方に基づき、語の (相対) 頻度や特定の構文パターンにより語の重要度を計量してきた [28, 29]。本研究ではこれを発展させた。

**[語彙の計量モデル]** 文書集合中の単語の意味を「文脈を共有する語の分布＝共起語分布ベクトル」などの統計的計量として扱うことを検討した。上記汎用連想計算エンジンを用いて語彙の計量モデルの計算手法を開発した。

**[語彙分析手法]** この計量モデルを使って、文書集合中での語の「話題性」の統計的な評価法を考案した。さらに、複数の語が複合することにより話題性が格段に強まる語が「重要複合語」であると考え、それらの自動抽出法を提案した。計算コストの高い共起関係解析に汎用連想エンジンを用いることにより、新聞１年分や百科事典などの大規模文書集合についても、ＰＣレベルの計算資源で上記の語彙分析が可能であることを実証した。また実際に話題性の高い語を集中的に提示することにより、検索支援機能が向上することを実証した。

**ベースとして用いる理論、技術**

我々はこれまで、階層的ベイズクラスタリング（ＨＢＣ）[31, 30] とそれに基づく文書連想検索 [43] を提案してきた。また、文書集合に含まれる話題の要約を、特徴的な単語群とそれらの共起関係グラフとして示す手法（特徴語グラフ） [35] を提案し、さらにこれらを組み合わせた検索インタフェース DualNAVI [42] を開発した。汎用連想計算エンジンの基本計算モデルとしては、これらの手法の核となっている各種の統計的類似性計算を一般化したものを用いる。

本研究開発は、文書クラスタリングや文書連想検索等の次世代情報検索技術の実現・評価に広く利用可能な、高速の汎用連想計算ソフトウェアを作成し、この分野の研究者に共通の研究・評価基盤を提供することを目的として、1999 年度より３年間の予定で開始した。

1999 年度は、汎用連想計算エンジン GETA(第１版) を作成し、その有望な応用分野である文書の自動分類技術

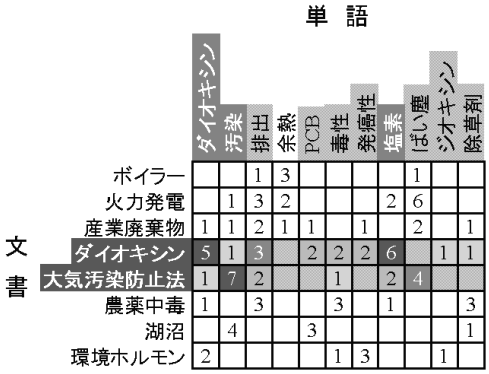


図 1. 文書の単語頻度WAMの例

と語彙の重要度を測る指標について、調査研究を行った。GETA は、複数の統計的計量の実装・差し替えが容易であり、C ／ Perl インタフェースを用いて高度な検索機能を簡便に実現できる。GETA の活用例として、評価用文書連想検索 GUI を実装し、各種統計的計量の定量的な比較評価が可能な実験環境を提供した。

2000 年度は、汎用連想計算エンジンの開発を継続し、特定計量についての最適化と、さらに大規模な文書集合への適用を考慮した分散 (分割) 処理方法について検討する。また、このエンジンを利用することで初めて実用性を持つと期待される動的クラスタリング手法等を実装し、有効性を確認する。また、これらの研究成果をフィードバックして、汎用連想計算エンジンの改良を図った。

2001 年度は、さらなるスケーラビリティーを達成するため高性能計算機として一般化しつつある PC クラスタ上で動作する分散処理方式を設計し、任意規模の PC クラスタで利用可能なＧＥＴＡ（第３版）を開発した。これにより 1,000 万件規模の文書コーパスへ適用することができるようになった。

## 2 汎用連想計算エンジン GETA の開発とその連想検索への応用

GETA と Perl モジュールを併用することにより，連想計算を用いた研究や様々な類似性尺度の比較・評価実験が簡単に行なえる．このようなツールは過去にも例がないため，情報検索分野の研究者に幅広く利用されると期待できる．

### 2.1 汎用連想計算エンジン GETA

汎用連想計算エンジン GETA (Generic Engine for Transposable Association) は，疎な行列を効率的に扱うための C 言語ライブラリおよび支援ツールから構成される．GETA は，行や列に固有の属性を付加した，**WAM** (Word-Article Matrix) というデータ構造を使用しており，WAM に対する様々な操作を C 言語ライブラリとして提供している．

文書検索に WAM を利用する場合は，行に文書，列に単語を割り当てる．行は文書の単語ベクトル (その文書にどの単語が含まれているか) を表現し，列は単語の文書ベクトル (その単語がどの文書に含まれるか) を表現する．このような行列をそのまま計算機上に実現すると膨大な記憶容量 (新聞 1 年分で約 100G バイト) が必要となるが，WAM を利用すれば約 $\frac{1}{1000}$ の記憶容量で実現できる．

WAM にアクセスすることにより，TF (term frequency，全文書中での索引語の出現数) や IDF (inverse document

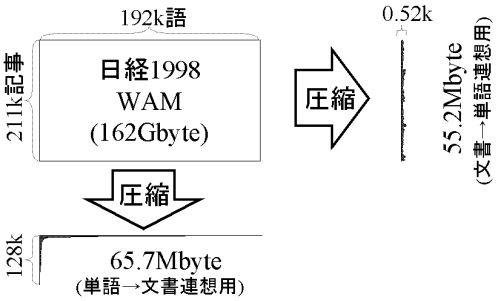


図 2. WAMの圧縮

frequency，索引語が出現する文書数の逆数)，単語間や文書間の類似度が計算できる．また，類似性尺度をユーザが任意に変更でき，種々の尺度の比較・評価実験が可能である．

### 2.2 連想検索の実装と評価

#### 2.2.1 Perl モジュールの作成

汎用連想計算エンジン GETA を用いると，計算コストの高い類似度計算を高速に実行でき，従来は実現が困難であった文書連想検索を簡単に実現できる．しかし，GETA を利用して連想計算を用いた研究・実験をするためには，高度な C 言語プログラミング技術が要求されるため，研究の効率上も，ツールとして公開していく上でも，大きな障害となる．このような障害を取り除くために，言語処理研究者の間で広く用いられている Perl 言語から使用できる Perl モジュールを作成した．Perl から C のコードを呼び出す方法としては，言語処理ツールでの活用実績等を調査した結果，h2xs と xsubpp というツールの組み合わせ (XS と呼ばれる) が最適であると判断し，それを利用した．Perl モジュールの実装によって，C 言語に精通していない言語処理研究者でも，連想計算を用いた研究や様々な類似性尺度の比較・評価実験を簡単に行なえる．

類似度尺度を変更する場合も，単に変数を変更するだけで簡単に行なうことができる (例えば，`wam::WT_SMART` を `wam::WT_LOG` に変更)．このように簡単に連想計算の評価実験を行なえるツールは過去に例がないので，研究者にとって有用なツールとして，広く利用されていくことが期待される．

#### 2.2.2 連想検索 GUI の開発

Perl モジュールを利用して連想検索を行なう GUI を開発した．本 GUI は，Perl で記述した CGI プログラムとして実装し，Web ブラウザからアクセスできるようにした．

GUI 上には，**単語連想検索**および**文書連想検索**を実装した．単語連想検索では，単語群を入力として，それらと関連する文書群を検索する．文書連想検索では，文書群を入力として，それらと関連する文書群を検索する．文書連想検索は，検索要求と関連する文書をあらかじめ持っている場合や，キーワード検索などで検索要求と関連する文書を一つでも見つけた場合に有効である．また，GUI 上には**適合性フィードバック**も採用し，得られた検索結果をもとに連想検索を繰り返し実行することによって，ユーザが必要とする情報を精度良く検索することを可能とした．

図 3 は，検索質問「フェラーリ」からの単語連想検索では 1 文書しか得られなかったため，その 1 文書から文書連想検索を行なった結果である．初期検索で得られた文書数が少ない場合，キーワード検索では検索語を追加する必要があるが，文書連想検索では検索された文書と関連する文書を検索できるため，検索語の追加は不要である．文書連想検索で得られた文書を見てみると，最初の検索質問「フェラーリ」に関連する文書 (自動車関係の文書) が上位に出現しており，文書連想検索の有効性が確認できる．

#### 2.2.3 類似性尺度の比較

GETA と Perl モジュールを用いて，簡単に類似性尺度を実装できることを活用して、代表的な類似性尺度の比較実験およびその評価を行なった．実験に使用した類似性尺度は以下の 3 種類である。

(1) ヒットする検索単語数を用いた方法 (`WT_HITS`)
(2) tf*idf 法 (`WT_TFIDF`)
(3) Singhal[39] の方法 (`WT_SMART`)

Perl モジュールを利用すれば，変数を変更するだけで簡単に類似性尺度の差し替えができる．毎日新聞の'94 年と'95 年 (約 21 万記事) を用いた検索評価実験を行ない，以下の結果を得た．

| 類似性尺度 | R-Precision |
|---|---|
| `WT_HITS` | 0.185 |
| `WT_TFIDF` | 0.195 |
| `WT_SMART` | 0.354 |

評価には R-Precision と呼ばれる評価尺度を用いた．実験の結果，一般の検索システムでよく使用される tf*idf 法よりも，Singhal の方法の方が優れていることが確認できた．また，GETA は検索対象文書の差し替えも簡単にできるため，様々な条件下での評価実験が可能である．

### 2.3 分散型連想計算

大規模なコーパスを対象として文書検索を行う場合に，分割したコーパスを複数の CPU に分散し，それぞれの CPU で連想計算を行った結果を統合する手法が考えられる (以下, 分散文書連想方式)．この手法は，コーパスを分割しなければ主記憶 (仮想記憶) に入り切らない様な巨大なコーパスを扱わなければならい場合等に特に有効である．すなわち，仮想記憶で扱えないサイズの問題を，補助記憶装置を利用して扱える様プログラミングすることは一般に困難である．また，そうできたとしても，補助記憶装置は遅い．更に，副次的な効果として，検索作業が複数の CPU で分散して並列に行われるため，検索の高速化を期待することもできる．

#### 2.3.1 複数の分散連想計算の管理

一般に，文書集合 $q$ とある単語 $t$ を与えられて，$t$ の $q$ での重要度を計算するためには，$q$ の要素で $t$ を含むベクトルにアクセスする必要がある．そして，文書連想計算を行う場合には膨大な量のベクトルにアクセスする場合が多い．アクセスする可能性のあるベクトルは計算しようとしている計算機上の，できる限り一次記憶 (主記憶) 装置上に置きたい．これらのデータが二次 (補助記憶) 装置上に追い出された場合，いちじるしい性能の低下を招くことがあるためである．また，一部の OS には，その様な巨大なデータを一度に扱えないという制約があり，その場合，実装レベルでの制約回避が必要なこともある．

$C$ の計算機上での表現が主記憶に入らない場合にその様な文書集合を扱う方法のひとつとして，$C$ を分割し，別々の計算機でそれぞれの部分に対しての $q$ に現れる単語 $t$ の重要度を計算し，最後にそれらを合成することで $q'$ を計算する，という方法が考えられる．

#### 2.3.2 確率的動作モード

$n$ 個の結果を要求する検索の場合，コーパスを分割した場合と，分割しない場合とで同一の結果を得るためには，分割した各ノードにおいて上位 $n$ 個の結果を返せば良い．そうするとことで，あるノードに全ての正解 (コーパスを分割せずに取り扱うシステムが返す答え) が集中した場合でも同一の結果を得ることができる．この場合，通信量は分

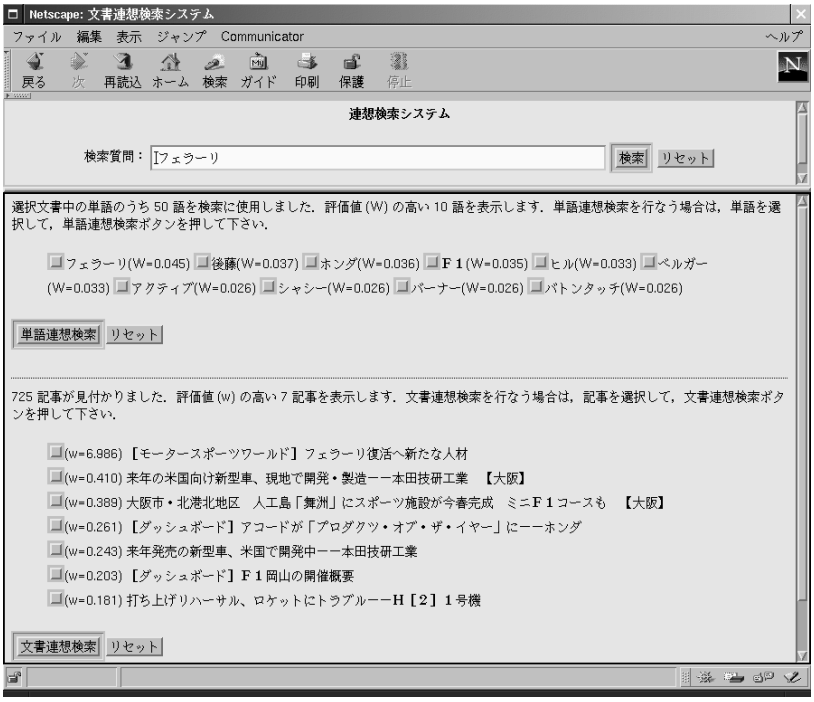


図 3. 文書連想検索の実行例

割数を $\ell$ として $n\ell$ となる．

コーパスに含まれる文章の数が十分大きいときに，$n$ 個の正解を $\ell$ 個のノードにランダムに割り当てると，あるノードにちょうど $k$ 個の正解が割り当てられる確率は，$p = \frac{1}{\ell}$ として，ほぼ ${}_nC_k p^k (1-p)^{n-k}$ となる (2 項分布)．さて，あるノードひとつに着目した時に，そのノードに $m$ 個以上の正解が割り当てられる確率 $q$ は，その累積確率を 1 から引いて，

$$q = 1 - \sum_{k=0}^{m} {}_nC_k p^k (1-p)^{n-k}$$

として得られる．ノードを特定せずに，どれかのノードに $m$ 個以上の正解が割り当てられる確率 $e$ は直接計算するのは困難だが，各ノードでの事象が独立では無いことから，$\ell$ 個のノードがある場合には その $\ell$ 倍以下，すなわち $e \leq \ell q$ であることは容易に分かる．

いま，$n$ 個の文章が要求されている時に，各ノードが上位 $m(< n)$ 個の文章のみを返すようにしたとする．この時，通信量は $m\ell$ となり，通信量を削減することが可能となる．この場合，どれかのノードに $m$ 個以上の正解が割り当てられると，この検索シテムは正解を返せなくなる (もしくは，正解の確信が持てなくなることがある，後述)．その確率は先に計算した $e$ 以下である．

そこで，$e$ が利用者の要求する値以下になる様に $m$ を決めれば良い．代表的な $e$, $n$, $l$ に対して，$m$ をいくらにすれば良いかという値を表 5 に示す．この表の簡単な見方について簡単に説明する．例えば，この表で最も右下の桝目に 28 とあるのは，“128 分割にて運用し，要求文書数が 1000 文書である場合に，各ノードがそのノードで検索した文書の上位 28 個だけを送れば (都合，集約ノードには 3584 文書が送られる)，いずれかのノードに 28 個より多くの正解が集中することが原因で，正解を取りこぼす確率 (誤り率) を 1/1000000 以下にすることができる，” ということである．ちなみに，誤り率を 0 にするには，各ノードが 1000 文書送らなければならず，都合，集約ノードは 128000 文書を受け取ることになる．これは，3584 の約 36 倍にもなる (1000/28 という計算でも同じ結果になる)．

#### 2.3.3 TCP/IP 対応の通信モジュール

TCP/IP はストリーム型の通信路であり，パケット指向ではないため，送信側ではパケットを書き込むだけであるが，受信側ではパケットの境界を検出しなければならない．実際にはパケットの構造を知らないでこれを行うことができればモジュール分けが可能になり，保守性も向上するので，実装では，パケットサイズをまず書き込み，それに続いてパケットを書き込むことでこの問題を解決している．

なお，UDP はパケット指向の通信路を提供するが，信頼性が無いので使用しなかった．また，一部の BSD 系 OS で実装されている SOCK_SEQPACKET はまだ広く使用できる段階ではないので，この使用も今回は見送った．

#### 2.3.4 分散型連想計算の評価実験

今回の実験では，連想エンジン部分の性能を見ることに重点を置いたため，dwsh のみの性能を測定した．連想検索システムとしての検索結果の整形，タイトル表示などにかかわる部分については測定していない．実験は，分割無し，1 ～ 7 分割の 8 通りについて，また，誤り率 1/1000000 の確率的動作の ON/OFF の 2 通りについて行った．分割した部分コーパスは dwsh を実行しているノード (計算機) 以外のノードに置き，ノード間は TCP/IP/Ethernet(100BaseTX, スイッチングハブ) で接続した．dwsh と各 wsh の間の通信プロトコルは TCP/IP 上に構築した独自プロトコルを用いている．

単語集合および文書集合の分割は，単語または文書を頻度順に並べ，その順位を分割数で割った余りに基づいて各部分コーパスに割り振った．

コーパスには，毎日新聞 1 年分 (2000 年)，毎日新聞 2 年分 (1999, 2000 年)，毎日新聞 4 年分 (1997, 1998, 1999,

2000 年) の 4 種類を用い, 形態素解析器で解析後, 品詞が, 名詞, 動詞, 形容詞, 未登録語, のいずれか, かつ語頭の 1 文字がアルファベット, ひらがな, カタカナ, 漢字, のいずれかであるものを選んだ. ただし, 語全体がひらがな 1 文字または 2 文字で構成されているものは除いた. 形態素解析器には, juman と互換性のある形態素解析器を用いた. 表 6 に実験に用いたコーパスの諸元をまとめる.

分割してセットアップを行うと, 各ノードに割り当てられた検索用のインデクスのサイズは小さくなる. インデクスには名前表などのメタデータが付随するため, $n$ 分割しても, そのサイズは $\frac{1}{n}$ よる大きくなる. 表 7 に各コーパスを分割しなかった場合と, 7 分割した場合のインデクスのサイズを示す. なお, 括弧内の数字は, 分割無し版のインデクスのサイズを 7 で割った値に対する比である.

これから分かる様に, コーパスを分割することで, 各ノードで扱わなければならないデータのサイズを小さくすることが可能であり, 当初の目的である大規模コーパスを扱うことが可能となる. しかし, 分割を行うことで, 検索システムトータルの性能が極端に落ちたりした場合, 実用的には大規模コーパスが扱えないということにもなりかねないため, その性質を調べておく必要がある. そのため, 実験では, 分割を行った場合の性能についても測定を行った.

## 3 文書クラスタリングを用いた情報アクセス手法

### 3.1 インタラクションモデル

対話的な文書検索の多くは，関連度フィードバック (relevance feedback) という手法を用いて，ユーザとシステムが対話的に情報を交換しながら検索結果の向上を試みる．具体的には，検索結果の幾つかの文書に対してユーザが適合/不適合性の判定を行い，これらの判定を使ってシステムは検索要求を更新し新たな検索を行う．

関連度フィードバックによる検索精度向上の割合は，ユーザがシステムに与えた判定の数に大きく依存する [6]. 我々は，ユーザが効率良くできるだけ多くの適合文書を選べるようなインターフェイスについて研究してきた [15, 11]．本論文では，文書クラスタリングを利用して検索結果を自動分類表示することの有効性を調べる．実際には「カテゴリーバー表示」と「デンドログラム表示」の二つの表示法について評価した．

カテゴリーバー表示では，クラスタリングアルゴリズムを用いて検索結果から 3 個の主カテゴリを見つけ，各文書とそれら主カテゴリとの距離をカラーバー表示する．ユーザは各々のカテゴリに注目して検索結果を並べかえることもできる．この表示法は，Scatter/Gather [7] で提案されている表示法や，Evans 等によって用いられた表示法 [9] と似たものでり，適切な分類に注目することで多くの適合文書を効率良く集めることができる．

デンドログラム表示では，階層的クラスタリングの結果をそのまま表示する．類似している文書対はなるべく近い場所に表示されるため，ある適合文書を種にして，その文書に類似する文書群も芋づる式に見つけることができる．

本研究では，NTCIR-2 [1] での実験を介して，上記二つの表示法を評価した．実験では実際にユーザとシステム間の対話を記録しているため，再現率/精度だけではなくユーザの行動も考慮した評価を行った．

### 3.2 検索結果の表示法

本研究で評価した検索結果の表示法を説明する．

#### 3.2.1 ランキング表示

検索要求との適合度が高い順に検索結果をならべて表示する．多くの検索システムで用いられているため，本研究でも提案手法のベースラインとしてランキング表示を用いる．ユーザは以下の操作を行うことができる．

| | |
|---|---|
| AVISIT | 指定した文書のフルテキストを表示する |
| ASEL | 指定した文書 (複数可) に適合マークを付ける |
| AUNSEL | 適合マークをはずす |

#### 3.2.2 カテゴリーバー表示

検索結果の各文書には，カテゴリーへの所属の度合を示すカテゴーバーが付いている．文書の初期並びはランキング表示と同じ適合度の順である．図 4 にカテゴリーバーの例を示す．これは，検索トピック「XML を用いた自然言語処理に関する論文」に対する検索結果の一部分である．

タイトルの横にある RGB スペクトルがカテゴリーバーである．それぞれの色 (R:赤，G:緑，B:青) は，検索結果 (実験では上位 150 文書) を要約する 3 個の主カテゴリに相当する．システムは階層的クラスタリングアルゴリズム [31] を使って，検索結果の文書群を 3 個のクラスタに分割する．これら 3 個のクラスタを主カテゴリとみなす．次に，各文書と 3 個のクラスタ間の距離を計算し，正規化の後に RGB スペクトルに変換する．よって，各色が占める割合は，対応するカテゴリーへの所属度の強さの割合に対応している．ここで，ユーザは各カテゴリの代表語を見ることができる．

| | |
|---|---|
| VCAT | 選択したカテゴリの代表語を見る |

あるカテゴリに興味を持った場合，ユーザはそのカテゴリに注目して現在の検索結果を並べかえることができる．

| | |
|---|---|
| GCAT | 選択したカテゴリに注目して検索結果を並べかえる |

この並べかえにより，注目しているカテゴリを代表する文書が上位に集まる．現在並べかえの方法として，

1) 注目カテゴリへの所属度が他カテゴリへの所属度よりも大きい文書のみを集める．ただし順位は検索要求との適合度の順である．
2) 単純に注目カテゴリの色の長さでソートする．

の 2 種類が選択可能である．図 5 は，上記の例に対して赤カテゴリに注目して並べかえを行った結果である．並べかえの方法は 1. の方法を用いた．

このように，カテゴリを介したインタラクションにより，ユーザは効果的に検索結果を絞りこむことができる．

なお，カテゴリーバー表示では，この他にも前述の AVISIT, ASEL, AUNSEL コマンドが利用できる．

#### 3.2.3 デンドログラム表示

デンドログラム表示では，階層的クラスタリングアルゴリズムの適用結果をそのまま表示する．階層的クラスタリングアルゴリズムでは，まずクラスタリングの対象文書それぞれを別々のクラスタとして設定する．次に一番近いクラスタ対をマージする．このマージを繰りかえすと最終的には以下のような木ができあがる．この木のことをデンドログラムと呼ぶ．

図 6 で左側のデンドログラムは文書の順序を整列しないで描いた木で，ここでは多数の枝が交差していて文書間の類似性が見にくい．右側の図は交差をほぐして文書を並べかえたデンドログラムである．類似する文書はできるだけ近くに配置されている．

本研究では，デンドログラムの木構造は表示せず，並べかえ後のデンドログラム (図 6 右) における文書の順序のみをユーザに提示する．とはいっても，この順序そのものは有用であり多くの情報を含んでいる．例えば，あるユーザが文書 3 を適合文書として選んだ場合，すぐ隣りの文書 6 もおそらく適合文書である．なぜなら文書 3 と文書 6 は類似しているからである．このようにして，種文書の近くにある文書を見ることで，種文書に類似する多くの文書を見つけることができる．

似ている文書が近くに配置されることでタイトル間の類

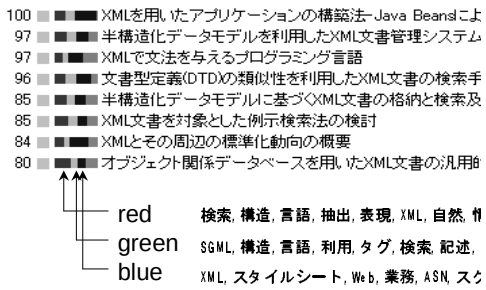


図 4. カテゴリーバーの例

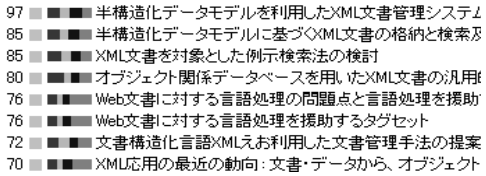


図 5. 赤カテゴリに注目して並べかえた

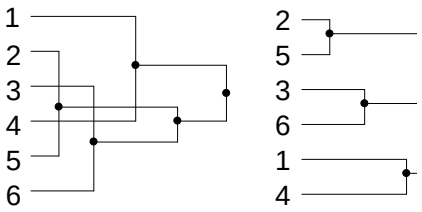


図 6. デンドログラム表示

**ranking**

84 [k]家庭科における学習が食生活に対する意識や価値観の形
a 84 生活価値観の変化に伴う新しい住要求に関する研究その2高
83 [k]浦東地区開発計画に伴う価値意識の変化に関する研究-E
:
82 東広島市における留学生の環境認識・評価に関する研究その
b 81 生活価値観の変化に伴う新しい住要求に関する研究その1研
81 バタン・ランゲージの方法による農村地域活性化のための生
:
77 [k]東京とロンドンとの空間構造と都市交通に関する比較研究
c 76 生活価値感の変化に伴う新しい住要求に関する研究：その4.
76 [k]職業特性の比較研究と価値志向の動向把握
:
71 在日外国人の住まい方に関する予備的研究
d 71 生活価値観の変化に伴う新しい住要求に関する研究その3パ
70 [k]「新・日本人の国民性調査」のための基礎的研究

**dendrogram**

62 過疎地域への転入定住者の実態と価値意識について山形県
a 84 生活価値観の変化に伴う新しい住要求に関する研究その2高
c 76 生活価値感の変化に伴う新しい住要求に関する研究：その4.
b 81 生活価値観の変化に伴う新しい住要求に関する研究その1研
d 71 生活価値観の変化に伴う新しい住要求に関する研究その3パ
80 東京の都市空間のイメージ特性に関する研究外国人との比
71 在日外国人の住まい方に関する予備的研究
67 アメリカに居住する日本人の住様式(第1報)-履床様式につい

図 7. ランキング vs. デンドログラム

似/差異といった一覧性も良くなり，ユーザはタイトルを見るだけでこれらの文書の関係をとらえやすくなる．例えば図 7 の例を見てみる．これは検索トピック「日本人の生活価値観の変化」に対する検索結果の一部分である．

今，文書 a, b, c, d に注目する．タイトルを見ればわかるように，これらは同じ著者による一連の論文である．ランキング表示では，適合度の値がばらついているため，これらの文書群が離れた位置に表示されているのに対し，デンドログラム表示では，まとまって近くに表示されている点に注目してほしい．よって，デンドログラム表示では文書 a, b, c, d がシリーズを成していることは一目瞭然である．

デンドログラム表示では，AVISIT, ASEL, AUNSEL コマンドが使用可能である．

### 3.3 実験環境

NTCIR-2 [1] の日本語検索に参加して，クラスタリングに基づく分類表示の評価を行った．以下は，NTCIR-2 に提出した結果の作成手順である．

1) システムは各トピックから初期検索要求を作る．具体的には，各トピックの <DESCRIPTION> と <NARRATIVE> フィールドからストップワードを除く全ての単語を抽出して検索タームとした．今回は <CONCEPT> フィールドは使わなかった．形態素解析プログラムには ANIMA [16] を，単語の重み付け法には Lt.Lnc 法 [19] を用いた．よって，検索システムはベクトル空間モデルに基づいていることになる．
2) システムは初期検索要求から 150 文書を検索し，以下のいずれかの表示法で各被験者に表示する．
   - ランキング表示 (3.2.1 節参照)
   - カテゴリーバー表示 (3.2.2 節参照)
   - デンドログラム表示 (3.2.3 節参照)
3) 各被験者は提示された検索結果から 15 分以内にできるだけ多くの適合文書をマークする．この間になされた操作は全て時刻付きで記録する．
4) 被験者がマークした適合文書をシステムにフィードバックして，システムは初期検索要求を更新する．具体的には，適合文書から上位 300 タームをフィードバックして改良 Rocchio 法により検索要求を更新した．改良 Rocchio 法のパラメータは $\alpha = 8$, $\beta = 16$, $\gamma = 0$ とした．つまり，負のフィードバックは行わなかった．
5) 更新した検索要求に基づいてシステムは再検索を行う．検索結果の 1,000 文書を評価対象として提出した．

図 8 に概略を示す．

7 人の被験者 (著者ら) が実験に参加した．

実験目的は，ベースラインのランキング表示とクラスタリングに基づく二つの表示法とを比較することである．

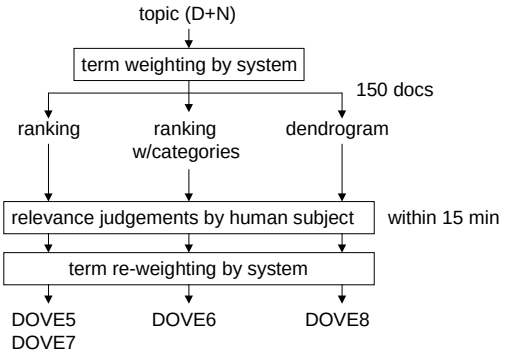


図 8. 対話的 run の概要

よって，各検索トピックでの一貫性を保つために，同じ被験者には二つの表示法について実験してもらった．一つはベースラインのランキング表示で，もう一つはカテゴリーバー表示，デンドログラム表示のいずれかである．ただし，二つの試行を行う順番が問題であるため，どちらを先に行なうかはランダムに決めた．更に，念のため，二つの試行の間には最短でも一週間の間隔をおいてもらった．まとめると，NTCIR-2 には以下の 4 つの対話的 run を提出した．

| | |
|---|---|
| DOVE5 | ランキング表示 (DOVE6 のベースライン) |
| DOVE6 | カテゴリーバー表示 |
| DOVE7 | ランキング表示 (DOVE8 のベースライン) |
| DOVE8 | デンドログラム表示 |

### 3.4 実験結果と考察

#### 3.4.1 総合結果

表 1 に平均精度 (average precision) を示す．ここで，S, A, B は適合性のレベルで，S は「特に適合」，A は「適合」，B は「部分的に適合」を意味する．

図からもわかるように，残念ながら，カテゴリーバー表示 (DOVE6)，デンンドログラム表示 (DOVE8) 共に，ベースラインのランキング表示を有為に上回ることができなかった．デンドログラム表示 (DOVE8) は，かろうじてベースラインを上回ったがその差は小さい．

図 9 は，平均精度の時間推移である．各時刻までにマークされた適合文書をフィードバックして得た平均精度をプロットしてある．ここでも，カテゴリーバー表示 (DOVE6) とデンドログラム表示 (DOVE8) の優位性は見てとれない．デンドログラム表示 (DOVE8) は，15 分近くになってやっとベースラインに追い付いているが，ほとんどの時刻においてベースラインを下回っている．

### 3.5 クラスタリングに関するまとめ

適合性フィードバックを効果的に行うための検索結果表示法として，二つの自動分類表示「カテゴリーバー表示」「デンドログラム表示」を提案し評価した．平均精度の観点からは両者の効果は認められなかったが，ユーザとの対話ログを調べた結果，特にデンドログラム表示の有効性が確認できた．検索結果をデンドログラム表示することにより，ユーザは類似する適合文書を効果的に集めることができた．カテゴリーバー表示に関しては，いずれの評価においても通常のランキング表示を上回ることがなかった．一つの原因として，インターフェイスが未熟でユーザが操作にとまどっていることが挙げられる．例えば，あるカテゴリーに注目した並べかえについて幾つかの手法をユーザに選ばせているが，明らかに複雑でわかりにくいインターフェイスである．今後は，カテゴリーバー表示のインターフェースを洗練化する予定である．

## 4 計量語彙モデルに関する調査研究

### 4.1 従来の研究

情報検索やターム抽出の分野でも，語の「話題性」や「分野代表性」(すなわち representativeness) を測るための指標が数多く提案されてきた [32]．語の重要度に関する指標は，歴史的には情報検索の分野で語の重み付けのために導入され，最も著名な例は tf-idf [38] であろう．idf は，全文書数 $N_{total}$ をある単語 $w$ が現れる文書数 $N(w)$ で割ったものの対数，tf は単語 $w$ の文書 $d$ 内での出現頻度 $f(w,d)$ であり，tf-idf は，これらの積として，$f(w,d) \times \log(N_{total}/N(W))$ で与えられる．さまざまな変形があるが，tf-idf の基本的な性質として，「単語がより多く，より少ない文書に偏って出現するほど大きくなる」ように設定される．上記文献には記述されていないが，この指標を特定の文書中での単語の重要度でなく，文書集合全体での単語の重要度を測る指標に拡張する自然な方法は，$f(w,d)$ を，w の全文書中での頻度 f(w) に置き換えることである．我々はこの平方を取った $f(w)1/2 \times log(N_{total}/N(W))$ を用いる．

他にも，注目する単語の，与えられた文書カテゴリごとの出現頻度の差異の偶然性を測り，偶然でない度合いが高いものを $\chi^2$ 検定で測る手法 [44] の他，隣り合う単語の共起の強さを様々な指標で測る手法が提案されている [41, 24, 27, 33]．

上記の各指標には，我々の目的に応用するには以下の問題があった：

(1) tf-idf (及びその類似手法) の精度は，経験上語の頻度の寄与が大きすぎ，「する」のような高頻度不用語が排除されにくい．
(2) 特定の語のカテゴリ間での分布の違いを比較する方法は，用途が限定される．
(3) 隣り合う単語の共起の強さを利用する手法では，１単語ごとの重要度が評価できない．
(4) 従来は重要／非重要を分ける閾値の設定が困難かつ恣意的になりがちであった．

このような問題の無い指標を構成する必要がある．

### 4.2 "representativeness" を測るための新指標

#### 4.2.1 基本方針

あるターム (単語または単語列) が"representative"であるとは，そのタームがある話題 (もしくは複数の話題群) を想起させてくれることを指す．この性質は，検索の結果得られた文書集合の内容を俯瞰し，新たにキーとなるタームを示唆する際に重要である．

このような性質を測る際の基本的な考え方として，"You shall know a word by the company it keeps" [25]. がある．本節では，これを数学的に言い替えることにより，タームの representativeness を測る指標を導入する．すなわち，

**W: ターム**
**D(W): W を含む文書すべての集合**
**D0 :全文書の集合**
**PD(W): D(W) における単語分布**
**P0 :D0 における単語分布**

とするとき，W の representativeness Rep(W) を，２つの分布 {PD(W), P0} の距離 Dist{PD(W), P0} に基づいて定義する．単語分布間の距離の計測方法としては，比較実験の結果，対数尤度比 (log-likelihood ratio) を用いた．

図 10 は，日経新聞 1996 年版の記事を用い，そこにあらわれるいくつかの語Wに対し，各語Wについて，D(W) の含む単語数 #D(W) を横軸に，Dist{PD(W), P0} を縦軸にプロットしたものである．図から見られるとおり，#D(W) が近いターム同士で比較すれば，たとえば「経済」

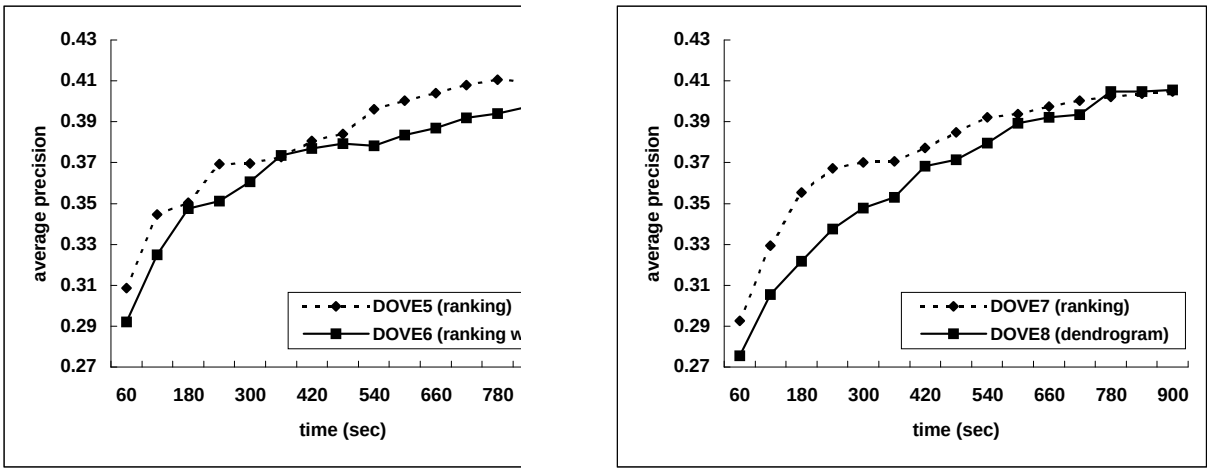


図 9. 平均精度の時間推移

| ID | other info | 平均精度 | |
|---|---|---|---|
| | | S+A | S+A+B |
| DOVE5 | ランキング表示 (DOVE6 のベースライン) | 0.4095 | 0.4020 |
| DOVE6 | カテゴリーバー表示 | 0.3996 | 0.3943 |
| DOVE7 | ランキング表示 (DIVE8 のベースライン) | 0.4052 | 0.3976 |
| DOVE8 | デンドログラム表示 | 0.4069 | 0.3891 |

表 1: 平均精度 (average precision)

は「する」,「電子」は「読み取る」より Dist {PD(W), P0} の値が高く直感と合致する．しかし，Dist{PD(W),P0} は，#D(W) が大きくなるにつれて増加するため，このままでは #D(W) が離れたターム同士の representativeness を適切に比較することはできない．実際，「電子」は「する」より Dist{PD(W), P0} の値が小さくなり，直感に合致しない。

#### 4.2.2 距離の正規化

そこで，さまざまな数の文書をランダムサンプリングし，その結果得られた文書集合 D に対して (#D, Dist{PD, P0}) を計算した。これらの点は，(0, 0) に始まり (#D0, 0) に終わる一つのなめらかな曲線により良く近似できる．以下，この曲線をベースライン曲線と呼ぶ．ベースライン曲線を指数関数を用いた近似関数 B で近似し，距離を B で正規化した値：

Rep(W) = DistPD(W), P0/B(#D(W))

により W の representativeness を定義する．

ランダムサンプリングした文書集合 D における DistPD, P0/B(#D) は，さまざまなコーパスにおいて，安定して平均 Avr がほぼ 1，標準偏差 $\sigma$ が 0.05 程度であり，最大値が Avr + 4 $\sigma$ を越えることはなかったので，Rep(W) の値が「意味のある値である」と判断するための閾値として，Avr + 4 $\sigma$ =1.20 を設けた．

### 4.3 実験結果

#### 4.3.1 新聞記事中のモノグラムに関する実験

日経新聞 1996 年版の記事中，総頻度が 3 以上の単語から 20,000 語を無作為抽出し，そのうちの 2,000 個を，検索内容の概観に現われることが「好ましい a」「どちらでもよい」「好ましくない　d」の 3 種類に人手で分類した．上記 20,000 語を何らかの方法でソートしたときに，各クラスに分類された語の，先頭から N 位までの累積出現頻度グラフを比較する．比較の対象として，ランダムソート，頻度，全文書を対象とした tf-idf の変形版を用いた．

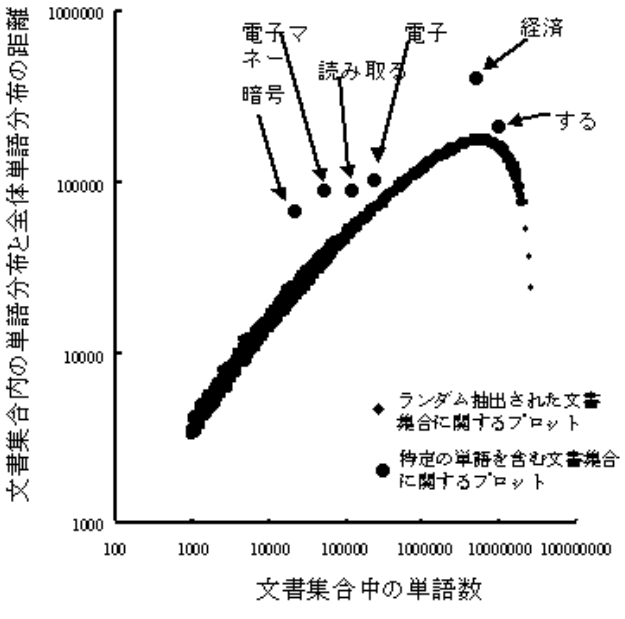


図 10. 文書集合中の単語の偏り

#### 4.3.2 結果

分類が"a"となったものの累積頻度を，ランダム，頻度，tf-idf，新指標のそれぞれを用いて比較した．ランダム＜頻度＜ tf-idf ＜新指標の順で「好ましい」と分類される語の優先順位を上げる力が強い．改善は (劇的ではないが) 有

為である．tf-idf は，重要語抽出においては上回ることがかなり困難な指標であるため [23]，充分肯定的な結果といえる．

分類が"d"となったものの累積頻度では新指標の選別能力の優位性がより際立っている．頻度と tf-idf はランダムな場合と変わらず，「不要語」特定能力が低い．

## 5 まとめ

本研究開発は、文書クラスタリングや文書連想検索等の次世代情報検索技術の実現・評価に広く利用可能な、高速の汎用連想計算ソフトウェアを作成し、この分野の研究者に共通の研究・評価基盤を提供することを目的として、1999 年度より３年間の予定で開始した。

1999 - 2000 年度は、各種の統計的計量に基づく連想計算の高速実行が可能で、使用する統計的計量を動的に変更できる汎用連想計算エンジンＧＥＴＡの開発を行ってきた。またＧＥＴＡの高速性能を損なわずに連想計算エンジンの基本機能を簡便に利用できる Perl インタフェースや高速クラスタリング・ライブラリ等の実験環境を提供した。

2001 年度は、電子化文書量の指数関数的増大に鑑み、1,000 万件規模の文書コーパスへの適用を目指して、高性能計算機として一般化しつつある PC クラスタ上で動作する分散処理方式を設計し、任意規模の PC クラスタで利用可能なＧＥＴＡ（第３版）を開発した。分散の度合（用いるＣＰＵの数）に応じた計算性能の測定も行い、文書量が増大すればする程、分散版ＧＥＴＡが必要かつ有効であることが明らかになった。

ＧＥＴＡを用いた大規模文書分析の調査研究としては、高速単語重要度計算プロトタイプシステムの開発を行い、統計的計量 (representativeness) を求める手法に基づく計量をツール化し、大規模な実験によりその有効性を検証することができた。

またＧＥＴＡの利用により初めて高速処理可能となった「文書クラスタリング」「語彙連鎖解析」「文書群を特徴づける単語群の抽出」等の基本的な文書分析手法をベースにして、「文書クラスタリング」を用いた新情報アクセス手法の調査研究の調査研究を行い、検索結果をクラスタリングして提示するインタフェースの効果を定量的に検証することができた。

## 謝辞

議論を通じて貴重なご意見をいただいた東京大学理学系研究科の辻井潤一教授、豊橋技術科学大学・知識情報工学系の増山繁教授に感謝致します。

## 参考文献

[1] *NTCIR Workshop 2: Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization*, 2001.

[2] I. J Aalbersberg. Incremental relevance feedback. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 11–22, 1992.

[3] J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 270–278, 1996.

[4] C. Buckley, M. Mitra, J. Walz, and C. Cardie. Using clustering and SuperConcepts within SMART: TREC 6. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, 1998.

[5] C. Buckley and G. Salton. Optimization of relevance feedback weights. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 351–357, 1995.

[6] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 292–300, 1994.

[7] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 318–329, 1992.

[8] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proc. of the Third Conference on Applied Natural Language Processing*, 1992.

[9] D. A. Evans, A. Huettner, Tong X., P. Jansen, and J. Bennett. Effectiveness of clustering in ad-hoc retrieval. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, 1999.

[10] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[11] M. Iwayama. Relevance feedback with a small number of relevance judgements: Incremental relevance feedvack vs. document clustering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 10–16, 2000.

[12] M. Iwayama and T. Tokunaga. Cluster-based text categorization: A comparison of category search strategies. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 273–280, 1995.

[13] M. Iwayama and T. Tokunaga. Hierarchical bayesian clustering for automatic text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1322–1327, 1995.

[14] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12, 1994.

[15] Y. Niwa, M. Iwayama, T. Hisamitsu, S. Nishioka, A. Takano, H. Sakurai, and O. Imaichi. Interactive document search with DualNAVI. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pp. 123–130, 1999.

[16] H. Sakurai and T. Hisamitsu. A data structure for fast lookup of grammatically connectable word pairs in japanese morphological analysis. In *International Conference on Computer Processing of Oriental Languages (ICCPOL'99)*, pp. 467–471, 1999.

[17] R. E. Schapire, Y. Singer, and A. Singhal. Boosting

and rocchio applied to text filtering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 215–223, 1998.

[18] H. Schütze and C. Silverstein. Projections for efficient document clustering. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997.

[19] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 21–29, 1996.

[20] A. Singhal, M. Mitra, and C. Buckley. Learning routing queries in a query zone. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 25–32, 1997.

[21] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[22] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973. 「クラスター分析とその応用」西田英郎監訳, 内田老鶴圃, 1988.

[23] S. A. Caraballo and E. Charniak. Determining the specificity of nouns from text. In *Proceedings of WVLC'99*, 1999.

[24] J. D. Cohen. Highlights: Language- and domain-independent automatic indexing terms for abstracting. *Journal of American Society for Information Science*, Vol. 46, No. 3, pp. 162–174, 1995.

[25] J. Firth. *A synopsis of linguistic theory 1930-1955*. Studies in Liguistic Analysis. Oxford, 1957.

[26] W. B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structure & Algorithms*. Prentice Hall, 1992.

[27] Katerina T. Frantzi, Sophia Ananiadou, and Junichi Tsujii. Extracting terminological expressions. *IPSJ SIG Notes*, Vol. 96-NL-112, pp. 83–88, 3 1996.

[28] Toru Hisamitsu and Yoshiki Niwa. Extraction of useful terms from parenthetical expressions by using simple rules and statistical measures - a comparative evaluation of bigram statistics -. In *Proceedings of COMPUTERM'98 (COLING-ACL'98 Workshop)*, pp. 36–42, 1998.

[29] Toru Hisamitsu, Yoshiki Niwa, and Yoshihiko Nitta. Acquisition of person names from newspaper articles by using lexical knowledge and co-occurence analysis. In *Proceedings of NLPRS'97*, pp. 329–334, 1997.

[30] M. Iwayama and T. Tokunaga. Cluster-based text categorization: A comparison of category search strategies. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 273–280, 1995.

[31] M. Iwayama and T. Tokunaga. Hierarchical bayesian clustering for automatic text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1322–1327, 1995.

[32] Kyo Kageura and Bin Umino. Methods of automatic term recognition. *TERMINOLOGY*, Vol. 3, No. 2, pp. 259–289, 1996.

[33] Hiroshi Nakagawa. Extraction of index words from manuals. In *Proceedings of RIAO'97*, pp. 598–611, 1997.

[34] Yoshiki Niwa, Makoto Iwayama, and Akihiko Takano. Interactive support of query refinement by dynamic word co-occurrence. In *Proceedings of ICCPOL'97*, pp. 383–386, 1997.

[35] Yoshiki Niwa, Shingo Nishioka, Makoto Iwayama, and Akihiko Takano. Topic graph generation for query navigation: Use of frequency classes for topic extraction. In *Proceedings of NLPRS'97*, pp. 95–100, 1997.

[36] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of COLING-94*, pp. 304–309, Kyoto, Japan, 1994.

[37] Tadashi Nomoto and Yuji Matsumoto. Data reliability and its effects on automatic abstracting. In *Proceedings of Fifth Workshop on Very Large Corpora*, pp. 113–126, 1997.

[38] Gerard Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, New York, NY, 1968.

[39] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of SIGIR'96*, pp. 21–29, 1996.

[40] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–49, 1999.

[41] 北研二, 小倉健太郎, 森元逞, 矢野米雄. 仕事量基準を用いたコーパスからの定型表現の自動抽出. 情報処理学会論文誌, Vol. 34, No. 9, pp. 1937–1943, 1993.

[42] 西岡慎吾, 丹羽芳樹, 岩山真, 高野明彦. 文献検索支援インタフェース *DualNAVI*. In *Proceedings of WISS'97*, pp. 43–48. 日本ソフトウェア科学会, 1997.

[43] 岩山真, 徳永健伸. 確率的クラスタリングを用いた文書連想検索. 自然言語処理, Vol. 5, No. 1, pp. 101–117, 1998.

[44] 長尾真, 水谷幹男, 池田浩之. 日本語文献における専門用語の自動抽出. 情処論文誌, Vol. 17, No. 2, pp. 110–117, 1976.

[45] 徳永健伸. 情報検索と言語処理. 言語と計算–5. 東京大学出版会, 1999.

[46] 丹羽芳樹. 動的な共起解析を用いた対話的文書検索支援. 情報処理学会・自然言語処理研究会報告, Vol. 96-NL-115, pp. 99–106, 9 1996.