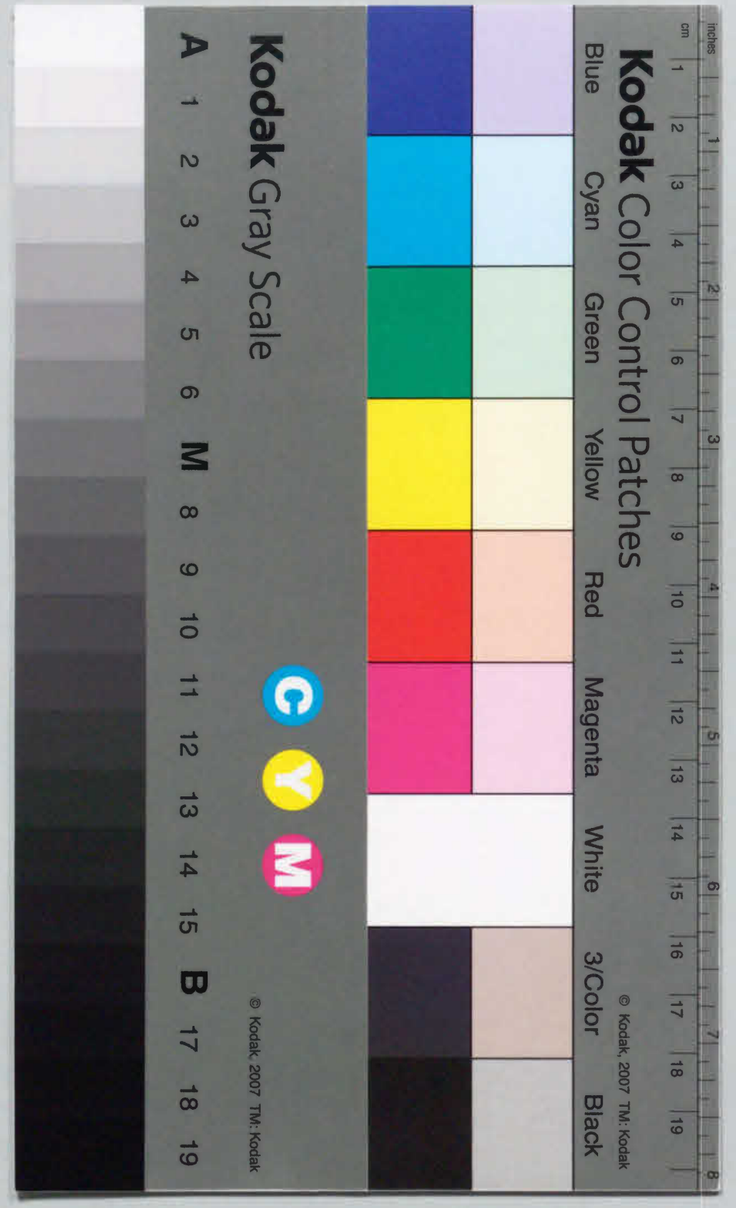

様式6

論　文　目　録

報告番号	甲　工 乙　工　第　193　号 工　修	氏名	佐々木 稔
学位論文題目	ベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究		

論文の目次

参考論文

主論文

1. “情報検索システムの統計的手法による特徴と精度の分析”, 佐々木 稔, 北 研二, 自然言語処理, Vol.8, No.1, pp.5-20 (2001).
2. “ランダム・プロジェクションによるベクトル空間情報検索モデルの次元削減”, 佐々木 稔, 北 研二, 自然言語処理, Vol.8, No.1, pp.85-100 (2001).

副論文

1. “Automatic Text Categorization based on Hierarchical Rules”, Minoru Sasaki, Kenji Kita, *Proc. of 5th International Conference on Soft Computing*, pp. 925–928, Kyushu Institute of Technology, JAPAN (Oct. 1998).
2. “Rule-Based Text Categorization Using Hierarchical Categories”, Minoru Sasaki, Kenji Kita, 1998 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2827–2830, San Diego, California, USA (Oct. 1998).

様式 7

論 文 内 容 要 旨

報告番号	(甲 工) 乙 工 第 **193** 号 工 修	氏名	佐 々 木 稔
学位論文題目	ベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究		

本論文は，情報検索手法の検索精度向上に関する研究として，情報検索システムに用いられた手法と検索精度に存在する関係の調査と概念ベクトルを用いることにより効率的に次元圧縮を可能とする，情報検索における新しい次元圧縮手法に関する研究の成果をまとめたものであり，以下の 6 章により構成される．

第 1 章では，緒論として，情報検索の歴史的背景を述べると共に，本研究の目的ならびにその工学上の意義を述べることで，本研究の意義及び位置付けを明確にする．

第 2 章では，情報検索システムの中でよく使われている検索モデルのひとつであるベクトル空間モデルを中心に，現在までに行われてきた単語の意味や共起関係などの情報を用いて検索を行う手法や，ベクトル空間の構造を利用してベクトルの次元を圧縮する手法として有効な，LSI (Latent Semantic Indexing) について説明する．

第 3 章では，IREX ワークショップにおける IR の本試験の結果，および，参加したすべての情報検索システムについてのアンケートを基に，平均適合率，再現率・適合率曲線を直線回帰させた傾きと切片が，情報検索システムに用いられた手法とどのような相関関係をもっているのかを調査し，それぞれの手法がシステムの性能に与える影響の大きさを示す．

第 4 章では，LSI の問題点を解決するために，ランダム・プロジェクションを用いた情報検索モデルを構築し，情報検索における次元圧縮手法として，ランダム・プロジェクションの有効性を確認する．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルに，文書の内容を表す概念ベクトルの利用し，これまで単語などが要素であったベクトルを文書の内容を要素とする低次元のベクトルに変換をするコンセプト・プロジェクションを提案する．

第 5 章では，提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法を提案する．このフィードバック手法は，判定評価の情報を初期検索要求に反映させるのではなく，コンセプト・プロジェクションの概念ベクトルに反映させているために、更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われ，フィードバック学習の影響が検索要求だけでなく検索対象にも反映できることを示す．

第 6 章で本研究で得られた諸成果の総括を行い，今後の研究課題について述べる．

ベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究

2001 年 3 月

佐々木　稔

ベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究

2001 年 3 月

佐々木　稔

内容梗概

本論文は，情報検索手法の検索精度向上に関する研究として，情報検索システムに用いられた手法と検索精度に存在する関係の調査と，概念ベクトルを用いることにより効率的に次元圧縮が可能となる，情報検索における新しい次元圧縮手法に関する研究の成果をまとめたものであり，以下の 6 章により構成される．

第 1 章では，緒論として，情報検索の歴史的背景を述べると共に，本研究の目的ならびにその工学上の意義を述べることで，本研究の意義及び位置付けを明確にする．

第 2 章では，情報検索システムの中でよく使われている検索モデルのひとつであるベクトル空間モデルを中心に，現在までに行われてきた単語の意味や共起関係などの情報を用いて検索を行う手法や，ベクトル空間の構造を利用してベクトルの次元を圧縮する手法として有効な，LSI (Latent Semantic Indexing) について説明する．

第 3 章では，IREX ワークショップにおける IR の本試験の結果，および，参加したすべての情報検索システムについてのアンケートを基に，平均適合率，再現率・適合率曲線を直線回帰させた傾きと切片が，情報検索システムに用いられた手法とどのような相関関係をもっているのかを調査し，それぞれの手法がシステムの性能に与える影響の大きさを示す．

第 4 章では，LSI の問題点を解決するために，ランダム・プロジェクションを用いた情報検索モデルを構築し，情報検索における次元圧縮手法として，ランダム・プロジェクションの有効性を確認する．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルに，文書の内容を表す概念ベクトルの利用し，これまで単語などが要素であったベクトルを文書の内容を要素とする低次元のベクトルに変換をするコンセプト・プロジェクションを提案する．

第 5 章では，提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法を提案する．このフィードバック手法は，判定評価の情報を初期検索要求に反映させるのではなく，コンセプト・プロジェクションの概念ベクトルに反映させているために、更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われ，フィードバック学習の影響が検索要求だけでなく検索対象にも反映できることを示す．

第 6 章で本研究で得られた諸成果の総括を行い，今後の研究課題について述べる．

関連発表論文

【主論文】

1) 佐々木 稔, 北 研二, “情報検索システムの統計的手法による特徴と精度の分析”, 自然言語処理, Vol.8, No.1, pp.5-20 (2001).
2) 佐々木 稔, 北 研二, “ランダム・プロジェクションによるベクトル空間情報検索モデルの次元削減”, 自然言語処理, Vol.8, No.1, pp.85-100 (2001).

【副論文】

1) Minoru Sasaki, Kenji Kita, “Automatic Text Categorization based on Hierarchical Rules”, *Proc. of 5th International Conference on Soft Computing*, pp. 925–928, Kyushu Institute of Technology, JAPAN (Oct. 1998).
2) Minoru Sasaki, Kenji Kita, “Rule-Based Text Categorization Using Hierarchical Categories”, 1998 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2827–2830, San Diego, California, USA (Oct. 1998).

【研究会資料】

1) 北 研二, 佐々木 稔, “離散フーリエ変換を用いたベクトル空間モデルの次元削減”, 情報処理学会自然言語処理研究会, NL133-10, pp. 69–76, 1999.
2) 佐々木 稔, 北 研二, “ランダム・プロジェクションによるベクトル空間モデルの次元削減” 情報処理学会自然言語処理研究会, NL135-4, pp. 25–32, 2000.
3) 佐々木 稔, 獅々掘正幹, 北 研二, “コンセプト・プロジェクションにおける関連性フィードバックを用いた概念ベクトルの更新手法”, 情報処理学会自然言語処理研究会, NL140-6, pp. 39–46, 2000.

【講演報告】

1) Kenji Kita, Minoru Sasaki, "Automatic Acquisition of Probabilistic Dialogue Models", 4th International Conference on Soft Computing, pp. 925–928, 1996.
2) Kenji Kita, Minoru Sasaki, "mprovement of a Probabilistic CFG Using a Cluster-Based Language Modeling Technique", 4th International Conference on Soft Computing, pp. 929–932, 1996.
3) 佐々木 稔, 北 研二, "階層的な規則を用いた文書のクラスタリング", 言語処理学会第 4 回年次大会, pp. 536–239, 1998.
4) Kenji Kita, Minoru Sasaki, Tai Xiao Ying, "Rule-Based Hierarchical Document Categorization for the World Wide Web", Asia Pacific Web Conference(APWeb98), pp. 269–273, 1998.
5) 佐々木 稔, 北 研二, "IR システムの特徴と精度の統計的手法による評価", the Proceedings of the IREX workshop, pp. 23–28, 1999.
6) 佐々木 稔, 北 研二, "文書の重みづけ手法を用いた情報検索システム", the Proceedings of the IREX workshop, pp. 81–86, 1999.
7) 佐々木 稔, 北 研二, "ランダム・プロジェクションを用いた情報検索システム" 言語処理学会第 6 回年次大会, pp. 431–434, 2000.
8) Tai Xiao Ying, Minoru Sasaki, Kenji Kita, Yasuhito Tanaka, "Improvement of Vector Space Information Retrieval Model based on Supervised Learning", the Proceeding of the Fifth International Workshop on Information Retrieval with Asian Languages(IRAL2000), pp. 69–74, 2000.

目 次

図目次

表目次

第 1 章　緒論

近年，情報化の浸透とインターネットの爆発的な普及とともに，WWW を代表とするネットワーク上の大量の電子データを個人が取り扱えるようになった．このため，個人が取り扱う必要のある情報量は，その個人が持っている選択能力の限界を超える程，非常に大きなものになっている．特に，新聞記事やインターネットのホームページなど，内容の定まらない非常に大きなテキストデータの中から，自分に有用で必要な情報を取り出すことは，非常に困難である．

このような状況を反映し，計算機を利用して非常に膨大なデータから必要な情報を取り出す情報検索や，文書から情報を取り出すテキスト処理を行うことにより，大量なデータに対する人間の情報処理能力を支援する情報検索技術の研究が盛んに進められている．計算機を利用する利点には，近年の計算機の処理能力やハードディスクなどのような記憶媒体の記憶容量の増加により，新聞記事などの情報を計算機で扱うことのできる内部表現に効率よく変換し，様々な処理を可能とする情報の形式化が容易にできることが挙げられる．例えば，Yahoo や Lycos などを代表とする組織的なインデックスサイトにおける検索エンジンにおいては，さまざまなクラスタリング情報，コメント情報などが作成されており，必要な情報を検索する際の有力な手段となっている．

現在では，検索の対象となるものには画像や音声といった範囲まで幅広く広がっている．しかし，以前においての情報検索の検索対象には，主として書籍や学術論文などといったテキストが用いられており，それらの表題や抄録を収録したデータベースから検索することが中心的な課題であった．これに対する解決策のひとつとして，いかにユーザの検索要求に満足な検索結果を与える情報検索システムを構築するかが，これまでの研究において考えられてきた．すなわち，情報検索のためのアルゴリズムや効率的なデータ構造の選択など，工学的なシステムの構築である．また，このようなシステムの性能を改善するために，自然言語から検索性能を向上させ，完全に自動的な索引語 (ターム) の作成を行うなどの自然言語処理技術を利用することが考えられる．このようなことを行うひとつの目的は，文書からその内容をよく表す索引語や意味内容を取り出すことである．仮に，文書中に出現する語句のみを用いて検索を行ったとすると，語句の多義性が検索結果に悪影響を及ぼす可能性が，少なからず存在している場合がある．現在では，形態素解析や統語解析など

の自然言語処理技術も急速に進歩し，簡単にかつ，高速にこれらの技術を利用できるツールが用意されていることもあり，できるだけ多くの検索要求に対して満足な検索結果を得ることのできる新しい検索手法の考案が重要な課題となる．

第2章では，情報検索システムの中でよく使われている検索モデルのひとつであるベクトル空間モデル [37] を中心に，現在までに行われてきた情報検索手法の研究を紹介し，それぞれの長所，短所を明確にする．ベクトル空間モデルは，文書と検索要求を多次元空間ベクトルとして表現する方法である．基本的には，文書集合から索引語とするタームを取り出し，タームの頻度などの統計的な情報により，文書ベクトルを表現する．この際，タームに重みを加えることにより，ひとつの文書，あるいは文書全体に対するタームの特徴を目立たせることが可能である．このような重みを計算するために，IDF(Inverse Document Freqency)[8] などの重みづけ方法が数多く提案されている．また，文書と検索要求を比較する類似度の尺度として，内積や余弦 (cosine) がよく用いられている．この類似度計算により，類似度の高いものからランクづけを行い，ユーザに表示することができることもベクトル空間モデルの特徴のひとつである．

これまでに，第2章で述べるような情報抽出，検索技術に関する研究が盛んに行われ，数多くの優れた日本語情報検索システムが提案されてきた．このようなシステムを評価するために，日本語テストコレクションの整備も進み，個々の検索システムを容易に評価できるようになった [20]．さらに，IREX (Information Retrieval and Extraction Exercise) ワークショップが開催され，共通のデータベースやプラットフォームにおけるシステム評価を行うことも可能となった．このような場においては数多くのシステムが参加しているため，ふたつのシステム間の比較実験では実験回数が莫大となり，システム間の相違点が多くなり，直接的に何が精度向上の原因であるのかをとらえることが難しくなる．従って，すべての検索システムを対象としてシステムの構成要素を評価すると同時に，全体的なシステム対する検索精度を評価するシステム指向の評価方法が必要と考えられる．第3章では，IREX ワークショップにおける IR 課題の本試験の結果，および参加した各システムについての，参加者が回答したアンケート結果を参考にして，IR 課題におけるシステムの特徴と精度の関連性を独自の統計的な手法を用いて分析を行う．

ベクトル空間モデルを用いた検索システムを新聞記事などの大量の文書データに対して適用した場合，文書データ全体に存在するタームの数が非常に多くなるため，文書ベクトルは高い次元を持つようになる．しかし，ひとつの文書データに存在するタームの数は文

書データ全体のターム数に比べると非常に少なく，文書ベクトルは要素に 0 の多い，スパースなベクトルになる．このような文書ベクトルを用いて類似度を計算する際には，検索時間の増加や文書ベクトルを保存するために必要なメモリの量が大きな問題となる．このため，単語の意味や共起関係などの情報を用いたり，ベクトル空間の構造を利用してベクトルの次元を圧縮する研究が盛んに行われている．このようなベクトルの次元圧縮技術には，統計的なパターン認識技術や線形代数を用いた手法などが用いられている [13][23]．この中で，最も代表的な手法として，LSI (Latent Semantic Indexing) がある [9][12]．この手法は，文書・単語行列を特異値分解を用いて，低いランクの近似的な行列を求めるものであり，これを用いた検索システムは，次元圧縮を行わない検索モデルと比較して一般的に良い性能を示す．しかし，特異値分解に必要な計算量が大きいために，検索モデルを構築する時間が非常に長いことが問題となっている．

第 4章では，上記の問題を解決するベクトル空間モデルの次元圧縮手法である，ランダム・プロジェクション [3] を紹介する．ランダム・プロジェクションは，あらかじめ指定した数のベクトルとの内積を計算することで次元圧縮を行う手法である．これまでに報告されているランダム・プロジェクションを用いた研究には，VLSI(Very Large-Scale Integrated circuit) の設計問題への利用 [45] や次元圧縮後の行列の特性を理論的に述べたものがある [3][31]．しかし，これらの文献では，ランダム・プロジェクションの理論的な特性は示されているものの，情報検索における具体的な実験結果は報告されていない．そのため，情報検索に対するランダム・プロジェクションの有効性に疑問が残る．

このような疑問点を解決するために，ランダム・プロジェクションを用いた情報検索モデルを構築し，情報検索における次元圧縮手法として，ランダム・プロジェクションの有効性を検証する．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルに，文書の内容を表す概念ベクトル [10] の利用し，これまで単語などが要素であったベクトルを文書の内容を要素とする低次元のベクトルに変換をするコンセプト・プロジェクションを提案する．このコンセプト・プロジェクションを用いることにより，任意のベクトルを用いた検索性能と比較して，検索性能が改善されていることを示し，この次元圧縮手法の有効であることを検証する．

第 5章では，提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法について述べる．関連性フィードバックは検索結果の各文書が正解であるか，不正解であるかをユーザに判定させ，この判定評価の情報を用いて初

期検索要求に反映させる手法である．これに対し，提案するフィードバック手法は，判定評価の情報を初期検索要求に反映させるのではなく，コンセプト・プロジェクションの概念ベクトルに反映させている．これにより，更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われるため，フィードバック学習の影響が検索要求だけでなく検索対象にも反映させることができる．関連性フィードバックによる様々な概念ベクトルの更新手法を提案し，テストコレクションによる検索実験結果を示し，更新手法の比較を行う．

第6章では，本研究で得られた諸成果の統括を行い，今後の研究課題について述べる．

第 2 章 情報検索

2.1 緒言

近年，情報化が浸透してきた現在において，社会の中にさまざまな情報がさまざまな形をなして存在している．これらの情報の中から必要な情報を素早く取捨選択し，効率よく利用することは，最近となっては人間にとって極めて日常的な行為となり，日常生活を行う上において必要な行動のひとつになっている．このような状況を反映して，コンピュータを利用して人間の持つ情報処理能力を支援することがこれまで盛んに行われている．

これを実現するために，現実に存在する情報や潜在的に存在する情報を概念化し，コンピュータで利用可能な内部表現に変換することによって，そのデータを形式的に保存する必要がある．これは，大量に存在する情報をコンピュータを利用して蓄積する操作で，この操作は，蓄積された情報が近い将来のおいて必要であることを予想しているために行われる．このために，情報をコンピュータに蓄積する方法を工夫し，その情報を利用する目的にかなった取り出しやすい形に変換し，保存させることにより，その情報を人間が効率よく利用することができる．

しかし，利用しやすい形に変換せずに，データをできるだけそのままコンピュータに蓄積し，情報の解釈はデータの出力を受け取った人間に任せる方法も存在する．このように蓄積された情報から，ユーザが必要だと思われる情報を検索し，ユーザに提示することが，現在情報検索としてよく知られている．この場合，多くの情報処理システムでは，ユーザの必要な情報を見つけるためにキーワードをあらかじめ抽出するといったある程度の処理は必要であるが，あらかじめユーザの意図を考慮した情報に変換するものではない．このとき，検索質問は，ユーザがある目的を満足するために持つ問題，すなわち，情報要求を具体的に表現したもので，検索対象から必要な情報をより確実に得ることができるように選択したものである．この検索質問から適合した情報を，これまでに蓄積された情報の中から選択することになる．

本章では，先の流れに沿う形でこの情報検索の概要をベクトル空間モデルを用いた情報検索システムを中心にこれまでに提案された，検索対象となる文書や検索質問の表現方法，情報検索システムの基幹となる検索モデルやその評価方法などの基本的な手法について説明する．

2.2 文書とそのコンピュータによる表現

情報検索の目的は，ユーザにより与えられる検索質問に適合する文書を探し出すことであるが，これまでの情報検索システムでは，検索対象である文書の一部分だけを用いての検索を行うまでに留まっていた．このように，文書の一部分だけを用いて検索を行うシステムの例として，図書検索システムが挙げられる．図書館において検索の対象となる文書の中には，文献を識別するための指標となる図書の評題，著者名，発行年などの書誌情報が盛り込まれている．これらの書誌情報や文献に付随する内容などといった情報をコンピュータに蓄積することにより，検索対象をデータベース化している．このような場合，検索結果として本の内容すべてを端末から見ることはできない．このため，従来の図書検索は，このようなデータベース情報からの検索結果をもとに，書庫で確認することによって，はじめて本の内容すべてを見ることができる．

しかし，ユーザにとっては，書庫に行くことなく，検索結果をもとに文書の一部分を確認できることがより便利なものとなる．このことは，多くの図書検索システムでは書誌情報が使われているため，検索結果からの早急な文書確認を実現することが困難な状況であった．このような文書に関する書誌情報を用いるのではなく，文書全体の内容を用いる検索システムの構築が強く望まれていた．

近年は，記憶容量の増加や CPU の性能向上などコンピュータのハードウェア面における性能向上に伴い，文書全体をコンピュータに蓄積し，それを用いての検索，いわゆる全文検索が可能となった．全文検索を行うことができれば，先の例のように，図書の一部分だけを検索結果としてユーザに提供できる，というように，更なるユーザの要求を満たせる検索システム構築が可能となる．

文書全体をコンピュータに蓄積するからといっても，文書全体を検索単位として検索質問をひとつひとつヒューリスティックにマッチングしているのでは，効率があまりよくないばかりか，例えば検索質問に「プリンター」が与えられた場合,「スプリンター」などのような，その文字列に対してそのままマッチングする文字列も関連のある文書であると判定される可能性も存在する．このため，効率の良い検索を行い，よりユーザの検索質問に関連のある文書が検索されるようにするために，文書の内容や書誌情報などをコンピュータが認識できるような内部表現形式に変換する必要がある．このうち，書誌情報は比較的内部表現に形式化しやすく，本の分類・整理などにも使われている．しかし，文書の内容情報を形式化するのは書誌情報のように簡単にはいかず，より精度の高い情報検索を目指すためにはこの課題が非常に重要になってくる．

内容情報を形式化するためには，文書から語を分割し，抽出するといった自然言語による表現を用いて，それに含まれる意味を抽出する必要がある．このような処理は自然言語処理 (natural language processing) と呼ばれている．一般的に，文書からその内容をよく表していると考えられる語を抽出し，抽出された語の集合によって文書内容を表現する方法が現在よく行われている．このような語は索引語 (index term) と呼ばれ，文書中において意味を持つもののなかで最小の構成単位として用いられている．

索引語の種類をどのような単位で選択するかについては，それぞれのシステムによって大きく異なっている．索引語の抽出という点に関しては，人手によることも考えられるが，文書の数が多くなると人手による方法では手間がかかり，現実的ではなくなってくる．これにより，これまでに索引語を自動的に抽出する研究が数多く行なわれており，数々の索引語抽出手法が提案されてきた．その中で索引語として抽出されるものは，一般的に単語や複合語であることが多い．このような単語や複合語を用いる場合，英語などのような分かち書きされている文書に対しては，単語と単語の区切りが明確であるため，容易に索引語を抽出することができる．しかし，日本語や中国語のように分かち書きされていない言語においては，単語を索引語をして用いるのは非常に困難であった．近年，与えられた文に対して品詞ごとに分割する形態素解析 (morphological analysis) [27] などのような自然言語処理技術の進歩により，精度の良い単語分割を行うことが容易に実現可能となり，日本語でも単語を索引語とすることが多くシステムで見られるようになった．

形態素解析を用いて自動的に文を分割し，それによりできたすべての語を索引語として扱った場合，助詞や助動詞などといったひらがなの表記が目立つ．これらは文書の内容を直接表わす重要な語であるとは言えないため，このような直接的に検索に関係ないと思われる語はあらかじめ削除しておいた方が良いとされている．このように，文書の特徴を表さないような語は一般的に不要語として扱い，索引付けを行う前にストップワードと呼ばれる不要語リストに登録しておき，そのリストに含まれる語は索引語としないようにしている．

しかし，不要語となる単語と不要語とならない単語との組合せを考えた場合，複合語とすることで全く別の意味を持ち，文書の特徴を表すようになることもある．例えば,『不』,『名誉』という 2 つの単語がある.『不』は次にくるものを打ち消す働きがあり,『名誉』はそれ自体が単独で意味をなす単語である．しかし，これらが作る複合語『不名誉』は全く

意味が逆になる．また，索引語と十分なり得る単語でも，複合語を作ることで，その複合語が文書の特徴を更に顕著に表すこともある．例として,『感染』と『予防』という単語を考える．これらの単語は単独でも意味をなす単語であるが，これらの作る複合語『感染予防』はより意味が限定されて，より鮮明に文書の特徴を表すことができる．このような複合語の抽出に関しては，単語を用いた場合と比較して，検索により有効な索引語であることは容易に理解できる．しかし，これまでの所は接頭語や接尾語などとの結合が行われるのが一般的で，より有効な複合語を抽出するのはこれからの課題となっている．

2.3 文書内容の索引付け

文書からユーザが検索するために重要であると考えられる語を抽出する処理のことは，一般的に索引付け (indexing) と呼ばれている．索引付けは，文書中からその文書の特徴を明確に表す索引語を余すことなく抽出することが重要である．この中でも，検索を行うために重要な索引語の特徴として，その文書の特徴を顕著に表す索引語をもれなく取り出す特定性と，文書の内容を消してしまわないように索引語を余すことなく抽出する網羅性がある．

特定性を高くするには，特定の文書内容のみに現われ，他の文書には現われないような索引語を抽出すればよい．そうすれば，検索質問でその索引語が用いられると，その文書内容を持つ文書が検索されることになり，検索精度の向上が期待できる．しかし，このような語のみを用いた場合，検索質問においてこのような索引語が使われる可能性も低くなるため，逆にその文書が検索されにくくなるという問題が生じてしまう．

また，網羅性を高くするために，一般によく使われる語を索引語として用いた場合，今度は索引語がさまざまな文書内容を持つ文書について頻繁に用いられているため，検索質問でこのような索引語が使われれば，利用者が欲している文書内容とは無関係の文書までも数多く検索されてしまう可能性がある．このように，特定性と網羅性とはトレードオフの関係にあり，両者のバランスをうまくとるような索引付け手法の研究が重要な課題となっている [1].

索引付けをするにあたり，索引付けを人間が行うかコンピュータを用いて自動的に行うかという選択肢が考えられる．人手による索引付けは，文書の内容を人間が実際に読んで理解した上での索引付けであるため，正確さという点では非常に優れている．しかし，文書の数が増えるに従ってこの作業は現実的ではなくなり，また，文書を読んだ人間によって索引語の選択が大きく変わってくる可能性もある．このため，索引付けの一貫性を保つのは非常に困難であると言える．これに対し，コンピュータを用いて自動的に索引付けを行った場合，ひとつの文書を何度も索引付けプログラムに入力しても，全く同じ結果が得られるため，索引付けの一貫性は保たれている．しかし，コンピュータが文書の内容を理解して索引付けを行うわけではないために，人間が見たときに，意味をなさない索引語を抽出しているという可能性がある．このような問題点に関しては，長年にわたり数多く研究されており，人手による索引付けと比べても，劣らない程度の精度，あるいは多少上まわる精度が得られたと報告されている．

また，自動的に索引付けを行う場合に，索引語を抽出する際の基本単位をどのような大きさに設定するのかが重要な問題となる．形態素解析のような自然言語処理解析技術を利用して検索に必要な索引語を抽出すれば，検索・分類の精度が上がることが期待される．分類にはそれぞれの分野の専門用語が重要な要素となる場合が多いが，特徴素解析を用いても正しく専門用語が切り出せるとは限らない．例えば，テキスト中に「情報検索」という用語があるとすると，そのままで切り出されると便利なのであるが，実際には「情報」と「検索」のふたつの単語に分割されてしまい，ひとつの単語としてうまく切り出すことができない．そこで，テキスト分類や検索するためのキーワードとなる特徴素を抽出するために，単に形態素解析を行って形態素に分割するだけではなく，各形態素の意味的な役割を考慮する研究が行われている．すなわち，形態素となり得る可能性の高いフレーズ (名詞句) を抽出し，これらに対しクラスタリングを行うことが考えられる [29]．ここで注目すべき名詞句抽出方法をを以下に挙げる．

1) 名詞連続の抽出
   「情報処理」,「情報検索」のような接頭語，接尾語を含めた名詞の連続をひとつの名詞として抽出する．このような専門用語は通常，辞書には未登録であり，より分野に特有な名詞句が抽出されると期待でき，形態素解析の結果として変な形態素の抽出を行っていたとしても，名詞句としての範囲を定めるという点については比較的うまく抽出できる．
2) 動詞連用形の処理
   日本語の動詞に対応するようなものの抽出は対象とはしていないが,「ばらつき」や「絞り込み」などのような連用形で表現され，前後の状況などから名詞のように使われていると判断される場合にはこれらの語句を抽出する．

3) 名前の抽出
「ベイズの定理」や「ワーズの方法」などの名前を抽出する．このとき連体助詞の「の」も含めて抽出する．

4) 状態を示すような名詞の除外
「機械的」や「一定」,「類似」などのような状態を表すものが抜き出した名詞句の前後に接している場合には，これを除外する．ただし，名詞が連続して出現するような場合にはこれらの句は抽出する．たとえば,「数が一定」の「一定」は「数」の状態を表すものとして抽出しないが,「異常気象」の場合にはそのまま抽出する．

5) 分野性の無い名詞の除外
「こと」や「もの」のような文書中の構造や他の場所を指定したり，筆者の思考や心的状態を示したり，事象間の関係などを示すような特定分野にかかわらない名詞は抽出しない．

6) 連体詞的，相対的，副詞的なものの除外
「該」,「同」や「中」,「付近」,「以上」,「現在」など，連体詞的，相対的，副詞的なものは名詞句の一部としては扱わない．

7) 番号の除外
「カウント 3」や「センサ 4」などの名詞の連続したものの後にくる数字は取り除く．これらは前にくる名詞の単なる ID と考えられ，削除しても何の問題は無いと考えられるからである．しかし,「号」や「世」など特定の接辞とともに用いられる数字は，これを含めて抽出する．

8) アルファベット
「RIPPER」や「FM」のような 2 文字以上から成るアルファベット列は抽出する．1 文字の場合は記号である可能性が高いのでこの場合は抽出しない．

これらの処理によって抽出した名詞句をフレーズと定義し，これを索引語とする方法と，フレーズの中から単一の語からなる名詞句を取り除いたもの複合語と定義し，これを索引語とする方法がどれほどの精度であるか，以下に示す 7 つの手法について比較，検討を行っている [29].

1) 単漢字
上述のように文書中の単一漢字を取り出したものを特徴素とする．

2) 名詞単漢字

全漢字を取り出したとすると,「中」や「以上」などの分野の特徴とはあまり関係の無い単語も取り出されてしまうことになる．これらの語はどのようなクラスの文書中にも平均して出現すると考えられるので，これらの語を取り除いたとしてもそれほどクラスタリングに影響を及ぼすことはないと仮定する．まず，形態素解析を行って名詞単語だけを抽出し，抽出された名詞単語から全漢字を一文字単位に取り出したものを特徴素とする．

3) 漢字単語
名詞単漢字は一文字単位で漢字を抽出していたのであるが，それをさらに拡張して漢字のみからなる単語だけを特徴素として残しておく．すなわち, 形態素解析を行って名詞と判定されたものからひらがな語とカタカナ語を取り除いたものである．

4) 単語
漢字単語を特徴素としたのではカタカナ語やひらがな語を特徴素として抽出していないので，ひらがな語とカタカナ語を取り除かずに得られた名詞単語をそのまま特徴素として扱う．

5) 漢字 bigram
これは，テキスト中の漢字の 2 文字の連続したもので，単独に出現する漢字やカタカナ語，ひらがな語，アルファベットは取り扱わない．たとえば,「日本語テキスト分類」からは,「日本」,「本語」,「分類」を特徴素として抽出する．

6) 名詞句漢字 bigram
上の漢字 bigram では,「上述」や「一定」というような状態を表すような副詞的名詞，記述性名詞などの特定の分野の特徴とはあまり関係の無い単語も取り出されてしまうことになる．したがって，フレーズから名詞単漢字と同様に状態を表す名詞を除く漢字 bigram を抽出したものを特徴素として扱う．

7) 単語 bigram
フレーズを抽出し，それが 2 形態素以上からなる時，その中からすべての 2 連続形態素を特徴素として抽出する．たとえば,「日本語テキスト分類」は,「日本語」,「テキスト」,「分類」の 3 形態素に分割されるので，この場合,「日本語テキスト」と「テキスト分類」の 2 つを取り出してそれを特徴素とする．

これら 9 種類の特徴素を比較すると，単漢字を特徴素とする方法が最も分類精度が悪い．これに対し，もっとも認識率が良かったものは単語 bigram で，次いでフレーズが良い結果

が出たと報告されている [29].

これまで，文書から抽出した索引語の集合を得ることによって，文書とその内容を表現することを述べた．しかし，それぞれの索引語が文書においてどれほどの重要度を持っているかについては全く考慮していない．同じ文書から抽出された索引語でも，その文書内容に直接関わる索引語は，より重要度が高いということができる．たとえば,「エイズワクチン」,「ＨＩＶ」などは「室長」,「所長」,「判断」などの索引語と比較すると，文書の内容に大きく関わる重要な語であることがわかる．このように，単に索引語を抽出するだけではなく，それぞれの文書，もしくは文書全体に対して索引語の重要度を与えることでより有効な情報検索が行われると考えられる．

2.3.1 不要語リスト

自然言語には大きく分けて，それ自体で意味を持った，ある特定の概念を表した内容語 (content word)，語と語の間の関係を表す機能語 (function word) がある．内容語には名詞，動詞が中心となって含まれ，文書内容を特徴づける語として用いられるが，場合によっては索引語とした方がいいものもあり，そうしない方がよいものもある．例えば，漢数字『五』，『十二』などは名詞ではあるが，一般的に文書内容とは関連性がなく，ユーザが検索質問として用いられることは希であるために，索引語から削除した方がよいと考えられる．機能語には助詞，助動詞などがあるが，これらの語は文書の内容を特徴付けるには，あまり効果的であるとはいえない．

どのような語が文書を特徴付けるかを判断するのは非常に難しいが，どのような語が文書を特徴付けないかを判断するのは，先の例のように比較的容易である．機能語以外の内容語に当てはまる語に対しては，経験的に文書を特徴付けなくても実際の検索性能にはあまり効果がないと考えられる語もある．例えば,「する」や「ある」などの動詞や「こと」や「もの」などの代名詞がそれにあたる．多くの文書に頻繁に出現する索引語が出現するために，特定の文書内容を顕著に表している索引語の重要度が小さくなっている場合には，頻繁に出現する，いわゆる一般語と呼ばれる索引語は省略してもよいと考えられる．従って，索引付けを行う際には，先に述べた機能語などのような検索にあまり効果が期待できない語は不要語リストに登録し，あらかじめ索引語から削除した方がよい．これにより，索引語の総数を減少させることができ，記憶容量の削減，処理の効率化や高速化などのコンピュータの処理を軽減する効果を得ることができる．

不要語リストの具体的な定義の仕方はさまざまな情報検索システムによって異なっているが，一般的な検索システムは機能語と一般的な語を不要語としているものが多い．機能語は形態素解析を行った後に出力される，それぞれの語の品詞情報をもとに決めることができる．一般的な語については，文書全体に出現する語の頻度によって決めることが多く，頻度により一般的な語であるかを判定する際には，頻度にある閾値を定め，それ以上の頻度をもつ語に関して，不要語としているものが多い．

2.3.2 接辞処理

情報検索システムでは，ユーザの検索質問の内容と文書の内容を比較し，類似性の高い文書をユーザに提供する．ユーザの与える検索質問と検索対象である文書との比較を行う際には，文書の内容，検索質問の内容はともに索引語の集合で表されているため，それら索引語を正確に比較することで類似性を求めることが重要となる．このため，索引語を用いて検索質問と文書の内容を比較する場合には，同じ事柄や物などに対して異なった表現を用いている可能性があることに，注意する必要がある．例として,「worker」,「working」,「works」などはすべて異なった表現であるが，同じ「work」という語の意味を表すものである．このような，索引語に対する表記のゆれや語形の変化に対処するために，シソーラスと呼ばれるデータベース化された類義語の集合を利用することにより，このような単語の集合に対して表記を統一し，ひとつの索引語としてまとめる手法がよく用いられる．

分かち書きの習慣のない日本語においては，語と語の境界を明確に示すことが難しいので，語形の変化は動詞以外にはあまり意識されないが，英語などのような名詞や動詞などの語形がさまざまに変化する言語に関しては，語形の多様性も考えられる．このように，場合によって語の形 (語尾) が変化することがあるため，索引語どうしの正確な適合が要求される場合には大きな問題となる．このため，索引付けを行う際にはこのような語尾変化した語を 1 つにまとめた方が，索引語数の軽減により，検索効率の向上が期待できる．

接辞処理のアルゴリズムの基本は，あらかじめ用意された規則に従って接尾辞を削除し，語幹 (stem) を出力することである [30][32]．しかし，医学や科学などの分野では造語が多く，接頭辞も語の意味がなくならない限り削除の対象になることもあるが，一般には接頭辞は意味を逆転するなど，意味を変化させるものが多いので接尾辞のみを処理の対象とすることが多い．情報検索の分野においては，語基が基本的な意味を表し，接尾辞などは統語的な性質を表しているという考えに基づいて，このように接尾辞に対して処理が行われる．下の例では，接辞処理 (stemming) によって一番右の語形に正規化する．

- knives → knife + s → knife

- happiest → happy + est → happy
- loving → love + ing → love

さらに，英語は多品詞が多いために各単語の品詞を決定することが重要となる．

2.4 検索質問の表現

本節では，検索したい文書と比較するための検索質問 (query) の表現方法について述べる．ユーザが自分の検索したいことを表現する場合に最も自然な表現方法は，自然言語によって自分の要求を表現することである．しかし，人間が日常用いている自然言語には，表記のゆれなどのあいまいさや，その時々によっていろいろな省略や言い替えが存在している場合がある．従って，利用者の要求するものを自然言語で表現した場合，現在の自然言語処理においては，自然言語の意味までも忠実に解析し，意味的な内容を抽出することは非常に高度な技術を必要とする．

このような高度な技術を必要としないように，別の手法として，索引語の集合によって検索質問を表現することが考えれられる．この手法は，文書の索引付けにより得られた索引語と同等なものを検索質問として用いるもので，その目的は，先に説明したように，文書中における索引語の集合と検索質問を比較し，類似度を求めるために行われるものである．現在利用されている多くの情報検索システムは検索質問をこのような索引語の集合として入力するものが多く，文書中の索引語と同様に，検索質問の索引語にも重みを付与できるという利点もある．しかし，ユーザはあらかじめ文書内容に含まれる索引語の集合を知らないため，文書中の索引語との厳密な適合が必要不可欠となるといった問題点も指摘されている．

また，論理式を用いて検索質問の表現する方法がある．索引語の集合では，単に利用者が欲しい情報に関して関連のある語を索引語として並べているだけであるので，並べた索引語どうしの関係を表していない．このため，索引語どうしの関係を命題論理の演算子を用いて関係を明確に表した索引語の集合を検索質問として用いる．演算子には以下に述べるものがある．

- 2 項演算子 AND：演算子で結ばれた索引語の両方が同時に文書中に出現していなければならない．
- 2 項演算子 OR：演算子で結ばれた索引語のうちどちらか一方でも文書中に出現していればよい．

- 単項演算子 NOT：その索引語が文書中には出現していてはいけない．

さらに括弧を用いることで，よりユーザの要求に沿った複雑な論理式を組み立てることができる．しかし，論理式による検索質問の表現にもいくつかの問題が指摘されている．まず，通常の論理式では，それぞれの索引語が利用者にとってどれだけ重要であるかを考慮した重み付けをすることができない．すべての索引語は文書中に出現するか出現しないかのみの判断によって検索結果が決まる．例えば,「梅田にあるレストランで，できれば無国籍料理，でなければタイ料理の店に関する情報」を検索したいとする．索引語の集合によって検索質問を表した場合,「無国籍料理」に対して「タイ料理」より高い重みを与えることによって，利用者が望む料理の種類の順に優先度を付けることができる．上記で述べたような問題に対しては，解決されているわけではないが，命題論理をつかって検索質問を表した場合，このような優先度をつけた表現は非常に困難である．また，複雑な論理式が利用者にとって理解することが難しいという問題もある．

2.5 検索質問拡張

情報検索システムでは，文書とユーザからの検索質問の適合性を，文書内の索引語と検索質問内の索引語とを比較よることで得られる適合度によって判定を行う．この際，索引語間のマッチングには，字面での厳密な比較が必要である．しかし，一般的に言語は多義的であり，1 つの概念を表す言葉にも類義語が多く存在する.『人』という言葉を例に挙げると,『人』には,『人間』,『人類』など,『人』という言葉が持つ概念と同じ概念を持つ言葉が複数存在することは容易に理解できる．このような場合，検索質問を『人』として検索を行うと，検索システムは文書中に『人間』という索引語が含まれている場合，この文書は『人』という検索質問に適合しない文書であるとみなされてしまう．このような問題を解決するためには，以下の 2 つの方法が考えられる．

1) 同じ概念を表す表現全てを同一の記号に変換する．
2) 検索質問中に含まれる表現をそれと同じ概念を表す全ての表現の集合に置換する．

1) の方法は，文書，検索質問中の索引語の中で，同じような意味や内容を表す語をすべて同一の概念に変換し索引付けを行う．例えば,「火」,「炎」を同一概念として索引付けを行うときに，すべて@ＦＩＲＥのような概念を明確に示す記号に置き換えて，それを索引語とする [43]．一方，2) の方法は，ある 1 つの表現を同じ概念を持つ表現の集合に拡張するもので，これを検索質問に対して行う方法が，検索質問拡張 (query expansion) と呼ばれる

ものである．検索質問拡張の目的は，検索質問中の索引語と文書中の索引語の不一致を減少させ，検索洩れを少なくさせることである．

検索質問拡張は，なんらかの基準を設け，その基準で同じ概念を表す語のグループを作る．そして，検索質問中にあるグループに属している語があれば，その語が含まれるグループの全ての語が検索質問中に存在するものとして検索を行う．例えば，検索質問中に『人』という索引語があるとする．その場合,『人』と同じグループに属している『人間』,『人類』という語も検索質問中に存在しているとみなし，検索を行う．したがって，検索質問を拡張することで，この例のように文書中に『人間』という索引語しかない文書に対しても，高い関連性を示すことが可能となる．

検索質問拡張には，同じ概念の語を利用したものの他に語の関連性を用いたものもある．例えば,『人』に関連するものには,『アメリカ人』,『警備員』などがある．これらの語の概念は『人』が表す概念とは同一とは言えないが，なんらかの関連性があるとは言える．検索質問に『人』という索引語あるとすると,『人間』,『人類』という索引語が文書中に存在しなくても,『人』に関連する索引語があれば，この文書を検索することができる．このように，同一の概念を持っていなくても，その語に関連する語があれば，それらの語を用いて検索質問を拡張する．

このように，検索質問を拡張するためには，語の関係を判断する知識が必要となる．この知識はシソーラス (thesaurus) やオントロジー (ontology) とよばれる辞書に記述される．これらの辞書は，語句を意味によって分類配列し，各語句についての同義語，類義語，上位語，下位語，反義語などを記述した統制語用語集である．以下に，シソーラスの一例を示す．

- WordNet
- 分類語彙表
- EDR 概念辞書

WordNet は英語のシソーラス，分類語彙表，EDR 概念辞書は共に日本語のシソーラスである．図 2.1にシソーラスの例として分類語彙表の一部を示す．分類語彙表には，およそ 32600 語がこのような形式で階層分類されている．ただし，分類語彙表には専門用語が入っていないので，その分野特有の語には対応していない．シソーラスの構築は 大変手間のかかる作業であるため，個々の分野で人手で作ることは困難であり，固有名詞や分野固有の専門用語は変遷が激しく，ほとんど使わないものも数多く定義されている可能性がある．したがって，分野毎のシソーラスを自動構築することが望まれる．

1.1 抽象的関係
　1.100 こそあど
...
1.2 人間活動の主体
　1.200 われ，かれ .....
...
1.3 人間活動–精神および行為
　1.300 心
...
1.4 生産物および用具
　1.400 物品 — 金品，異物，現品，安物，名物，.....
　1.401 持ち物・売り物・みやげなど — 獲物，出土品，私物，忘れ物，.....
　....
　1.471 道路・橋 — 国道，地下道，線路，架け橋，.....

図 2.1 シソーラスの例 (分類語彙表の一部)([22] から引用)

2.6 ベクトル空間モデル

ベクトル空間モデル (vector space model) とは，文書や検索質問を多次元空間上のベクトルとして表現するものであり，これら 2 つのベクトルを比較するために類似度を計算し，検索質問に対する文書の適合度を計算するモデルである [37]．ベクトルの各次元には文書集合中に存在する索引語の数を割り当て，ベクトルの要素には出現する文書に対する索引語の重要度，あるいは，文書集合全体に対する索引語の重要度を重みとして数値化される．

m 個の索引語から成る n 個の文書集合が存在するとき，各ベクトルを並べて得られる $m \times n$ の索引語・文書行列が得られる．n 個の文書を表す各ベクトルはそれぞれ行列の列ベクトルを表し，文書ベクトルと呼ばれている．また，行列の行ベクトルは，索引語ベクトルと呼ばれ，対応する索引語がどの文書に出現するかを表したものである．この行列 A の要素 a_{ij} は，文書 j に出現する単語 i の頻度に，後述する様々な重みを加えた数値となり，その文書に対する索引語の重要度を表している．検索質問も同様に，索引語に重みを加えたベクトルとして表現することができ，検索質問ベクトルと文書ベクトルとの類似度を距離や内積などを用いて計算する．このように，情報検索の観点で重要なことは，検索

$m = 6$ の索引語:
T1: bak(e, ing)
T2: recipes
T3: bread
T4: cake
T5: pastr(y, ies)
T6: pie

$n = 5$ の文書のタイトル:
D1: How to Bake Bread Without Recipes
D2: The Classic Art of Viennese Pastry
D3: Numerical Recipes: The Art of Scientific Computing
D4: Breads, Pastries, Pies and Cakes: Quantity Baking Recipes
D5: Pastry: A Book of Best French Recipes

図 2.2 索引語と文書集合の例 ([5] から引用)

質問ベクトルと文書ベクトルとの間の類似性や相違性をモデル化するために，文書間の幾何学的な関係を得ることができる点である．

簡単な例として，図 2.2 に示す 6 個の索引語からなる 5 個の文書タイトルの集合から，6×5 の索引語・文書行列を得る様子を図 2.3 に示す．文書の内容は文書中に出現する索引語の頻度で表されるために，行列の要素は各文書ベクトルである列ベクトルのユークリッドノルムが 1 となるように正規化されている．

$$\|a_j\|_2 = 1 \quad (j = 1, \cdots, 5) \tag{2.1}$$

索引語の選択には，文書集合中のすべての単語を索引語とするだけでなく，検索に有用な索引語を用いることも可能である．図 2.3 の例では，料理 (cooking) に直接関連のある語句を索引語として選択し，索引語・文書行列を作っている．また，'to' や 'the' のような非常に一般的な単語は，文の係り受け構造を知る上では重要であるが，文書の内容を明確にするほどの重要さはほとんどない．このため，このような文書集合全体にわたり非常に高い頻度で用いられる語は，停止語リスト (stop words, stoplisting) と呼ばれる不要語辞書に登録され，索引付けを行う際にこのリストにある単語は索引語として用いられないようにしている．

正規化を行う前の 6×5 の索引語・文書行列 $\hat{A}$ を表し，その要素 $\hat{a}_{ij}$ は，文書タイトル j に出現する索引語 i の頻度とする．

$$\hat{A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

列ベクトルを単位ベクトルにした，6×5 の索引語・文書行列 A を表す．

$$A = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$

図 2.3 索引語・文書行列の例 ([5] から引用)

また，索引語・文書行列を作る際，索引語は通常，単語の語幹を用いている．これは，索引語が名詞が複数形になっていたり，動詞が進行形になっている場合には，名詞の単数形や動詞の原型と異なるものとして扱われてしまうために，文書にある索引語の正確な頻度分布を得ることが非常に困難になる．このため，単語の語幹を取り出す接尾語処理を用いて，語幹を索引語として扱う必要がある．図 2.3 の例では，'pastries' は 'pastry' と変換されて頻度が数えられ，'baking' は 'bake' と変換されて頻度が数えられている．

2.7 文書ベクトル

ベクトル空間モデルでは，文書中からその文書を特徴付ける索引語を余すことなく抽出することが，ベクトルを作成するための索引付けの主な役割である．さらに，抽出した索引語がその文書の内容にどの程度関係しているのかに従って，索引語の重要度として数値的な情報を索引語に与えることを索引語の重み付け (term weighting) という．索引語に対して重み付けを行うことにより，抽出した索引語に文書内容に関して重要度が与えられるために，より文書の内容を特定することができると考えられる．また，検索質問に対して

文書との類似度を計算する際は，各文書で重み付けされた索引語の重要度を用いることにより類似度が数値的に表されるために，検索質問に対する各文書の類似性を順序付けすることが可能になる．

特徴ベクトルを用いたクラスタリングでは，上に示したような特徴素を用いて，文書の特徴を損ねないような特徴ベクトルを作成する．特徴ベクトルの表現方法で一般的によく用いられているものは，文書中の特徴素を要素としたベクトル表現である [25]．たとえば，“*Speech and Speech Based Systems*” という文書を特徴ベクトルに変形すると次のように表現できる．

$$(\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\cdots)$$

ベクトル中の 1 が文書中の単語 *speech*，*based*，*systems* に対応し，文書中に存在しない単語は要素の値が 0 となっている．このベクトルは文書中に単語が存在するかどうかで要素の値が決定しているが，この表現方法では文書中の単語の頻度を全く無視することになる．これではひとつの文書中に何度も存在している単語，フレーズなどの特徴を捉えることができず，情報を損ねてしまう結果になりかねない．この問題を改善するために要素の値を 2 進表現ではなく，以下のように実数を用いて特徴が文書中に複数回存在しても対応できるようにする．

$$(\ 0,\ 0,\ 0,2.0,\ 0,\ 0,1.0,\ 0,1.0,\ 0,\ 0,\ 0,\cdots)$$

この例では単語が 1 回出現するごとに 1.0 を加えている．このようにそれぞれの属性に重みを加えた統計的な手法では，出現頻度ごとに要素の数を増やしていく．その場合，長い文書ほど同一の特徴素の頻度が多くなってしまい，短い文書との釣合が保てなくなる．そのために文書中に存在するすべての特徴素の数でひとつの特徴素の数を割ることにより，文書中に存在する特徴素の割合を求める．

ベクトルの成分や木のノードの表す数値，またはこれらを用いていない単独のスコアを計算するために，形態素や n-gram の統計を用いて文書中の索引語に重みをつける索引付けの方法が数多くある [8][14]．索引付けには，局所的重み，大域的重みと正規化と大きく 3 つの重み付けに分類することができ，索引語に対する重みは以下のようにこれらの重み付けの積で表される [4]．

$$d_{td} = W(d,t) = L_{td} \times G_t \times N_d \tag{2.2}$$

ここで，L_{td} は d 番目の文書に対する t 番目の索引語への重み，G_t は文書全体に対する y 番目の索引語への重み，また，N_j は文書 j に対して正規化を行う作用素を表す．局所的

表 2.1 局所的重み

名前	関数
二値 [36]	$\begin{cases} 1 & \text{if } f_{td} > 0 \\ 0 & \text{if } f_{td} = 0 \end{cases}$
頻度 [36]	f_{td}
対数 1[11]	$\begin{cases} 1 + \log f_{td} & \text{if } f_{td} > 0 \\ 0 & \text{if } f_{td} = 0 \end{cases}$
対数 2[16]	$\log(1 + f_{td})$
正規化対数 [7](a_j は文書 j 中にある索引語の平均頻度)	$\begin{cases} \frac{1+\log f_{td}}{1+\log a_d} & \text{if } f_{td} > 0 \\ 0 & \text{if } f_{td} = 0 \end{cases}$
正規化頻度 [35]	$\begin{cases} 0.5 + 0.5\frac{f_{td}}{\max_k f_{kd}} & \text{if } f_{td} > 0 \\ 0 & \text{if } f_{td} = 0 \end{cases}$
改良型正規化頻度 (K は $0 < K < 1$ の定数)[36]	$\begin{cases} 0.5 + 0.5\frac{f_{td}}{\max_k f_{kd}} & \text{if } f_{td} > 0 \\ 0 & \text{if } f_{td} = 0 \end{cases}$

重みは，一般的に一つの文書に対する各索引語の頻度に関する関数を表し，通常検索質問や与えられた文書に対して計算が行われる．大域的重みは，文書全体に対しての各索引語の頻度の関数と定義され，検索質問のような文書集合のデータ数が少ない場合には効果がなく，比較的大きな文書集合に対してこの重みが定義される．正規化は，文書間の文書の長さを一定に揃えるために行うものである．

局所的重みは，文書中に高い頻度で出現する索引語がその文書に直接的に関係がある語であるとして，非常に良い性能を示す重み付け手法である．簡単な局所的重みの手法として，単に文書 d 中の索引語 t の頻度 $\mathrm{TF}(d,t)$ を重みとするものがあり，次式のように重みを与える．

$$L_{td} = \mathrm{TF}(d,t) \tag{2.3}$$

この場合，索引語に名詞や動詞のような重要な索引語になりやすいものに関しては，非常に良い性能を示すが，自然言語に必ず存在している機能語 (助詞，前置詞，冠詞) も索引語として考慮している場合もこの値が高くなり，この重み付けだけでは識別能力はほとんど無くなってしまう可能性が高い．このような局所重みの例を表 2.1 に示す．

このように，頻度の高い索引語が重要であるとした場合，頻度の低い索引語は高い重み

が与えられない．しかし，頻度の低い索引語は一般的に複数の分野にわたって共起する可能性が低く，このような索引語が存在するだけでも，分野を特定することができる．このように，頻度の低いものに高い重みを与えるために，索引語の特定性を表す尺度として定義されたのが大域的重みである．この大域的重みで最もよく知られているものに，Inverse Document Frequency(IDF) がある．十分大量な文書数を N とし，この文書集合の中で項 t が含まれる文書数を $df(t)$ とすると，IDF の値 IDF(t) は次のように定義できる．

$$\mathrm{IDF}(t) = \log \frac{N}{df(t)} \tag{2.4}$$

これを用いて重み $W(d,t)$ を

$$W(d,t) = TF(d,t) \times IDF(t) \tag{2.5}$$

とすることで頻度のみ重みとするよりも良い動作をすることが示されている．

IDF はすべての文書の中で重みを計算したい項を含む文書の数を用いて計算している．そのため，項が一文中に数多く出現してもただ一回しか出現しなくても同一のものとして扱われる．したがって，これに重みをつけるようにひとつの文書中の計算したい項の数を全文書中のその項の数で割ったものが Weighted Inverse Document Frequency(WIDF) である．すなわち D を文書全体の集合とすると，

$$\begin{aligned} \mathrm{WIDF}(d,t) &= \frac{\mathrm{TF}(d,t)}{\sum_{i \in D} \mathrm{TF}(i,t)} \\ &= \frac{\mathrm{TF}(d,t)}{\mathrm{TF}(N,t)} \end{aligned} \tag{2.6}$$

となり，これを用いた重み $W(d,t)$ は次のようになる．

$$W(d,t) = \mathrm{WIDF}(d,t) \tag{2.7}$$

例として，表2.4にある文書について考えてみる．ここで表の中にある d_i はひとつの文書，また t_x，t_y は単語などの特徴素を表す項であり，その中の数字は文書中の特徴素の出現する頻度を表している．まず，単純な項の頻度を重みとする場合，表の中の数字がそのまま TF(d,t) の値となり，具体的には $\mathrm{TF}(d_1,t_x) = 2, \mathrm{TF}(d_2,t_x) = 50$ となる．次に，IDF は項 t_x，t_y が d_1 から d_5 までのどの文書中にも存在しているので，$\mathrm{IDF}(t_x), \mathrm{IDF}(t_y)$ はともに 0 となる．WIDF は，すべての文書に存在する項の頻度の和をまず計算し，ひとつの文書中の項の頻度を和で割るので，

$$\mathrm{WIDF}(d_2,t_x) = \frac{50}{2+50+3+2+4} = \frac{50}{61} \tag{2.8}$$

	d_1	d_2	d_3	d_4	d_5
⋮	⋮	⋮	⋮	⋮	⋮
t_x	2	50	3	2	4
⋮	⋮	⋮	⋮	⋮	⋮
t_y	3	2	3	2	3
⋮	⋮	⋮	⋮	⋮	⋮

図 2.4 文書と単語の関係

表 2.2 大域的重み

名前	関数
IDF[11][36]	$\log\left(\frac{N}{n_t}\right)$
確率的 IDF[16][36]	$\log\left(\frac{N-n_t}{n_t}\right)$
エントロピー [11]	$1+\sum_{j=1}^{N} \frac{\frac{f_{tj}}{F_t}\log\frac{f_{tj}}{F_t}}{\log N}$
GfIDF[11]	$\frac{F_t}{n_t}$
対数 GfIDF[8]	$\log\left(\frac{F_t}{n_t}+1\right)$
改良 GfIDF[8]	$\frac{F_t}{n_t}+1$
平方根 GfIDF[8]	$\sqrt{\frac{F_t}{n_t}-0.9}$
平均 [11]	$\frac{1}{\sqrt{\sum_{j=1}^{N} f_{tj}^2}}$
重みなし	1

となる．IDF と WIDF のどちらの方が優れているかを比較すると，文書どうしの比較ではなく，文書とクラスタとの比較でカテゴリを割り当てるという方法に限定しているが，WIDF が IDF より優れているという報告がなされている [44].

式 (2.6) にあるように，WIDF には反比例する項 TF(N,t) が存在している．この値が大きくなるほど WIDF の曲線は x 軸，y 軸にほとんど接したものになってしまい，この影響は TF(d,t) より大きくなり良い動作をしなくなると考えられる．このために WIDF にある反比例の項は使わず，WIDF の代わりにフェルマー型の曲線を用いた重み付けが提案されている [2]．フェルマー型の曲線とは，以下の式を満たすような曲線を表す．

$$F_a(t)^a + \mathrm{TF}(N,t)^a = MAX(\mathrm{TF})^a \tag{2.9}$$

ここで MAX(TF) は N 中で最も多く出現する項の回数，a は曲線次数である．この式を

表 2.3 正規化手法

名前	関数
余弦 [35]	$\frac{1}{\sqrt{\sum_{i=1}^{m}(G_i \times L_{id})^2}}$
Pivoted Unique[8][39]	$\frac{1}{(1-slope)+slope l_j}$
重みなし	1

用いた重み関数は次のようになる．

$$W_a(d,t) = \mathrm{TF}(d,t) \times F_a(t) \tag{2.10}$$

この重みの方が WIDF よりも良い結果を示し，一般的な頻度と形態素の重みとの対応がとれているとしている．しかし，比較的小さい文書に対しては，情報量が少なく，分類が比較的困難になるという指摘もなされている．

索引付けの三番目の要素である正規化は，文書集合に含まれている各文書の長さの相違を一致させるために用いられる．文書の長さが一定しない文書をベクトルで表現する場合には，文書ベクトルを正規化することが検索に対して有効である．よく用いられる正規化の手法を表 2.3 に示す．

ベクトル空間モデルで最もよく用いられる正規化手法は，ベクトルの余弦によるもので，以下のように表される．

$$N_j = \frac{1}{\sqrt{\sum_{i=1}^{m}(G_i \times L_{id})^2}} \tag{2.11}$$

これは，局所的重み，大域的重みによって重み付けされた文書ベクトルの大きさで文書ベクトルを割ったもので，これにより文書ベクトルの大きさが 1 に正規化される．

2.8 類似度計算

ベクトル空間モデルでは，文書集合に検索を行いたいユーザが，関連のある文書を見つけるために，検索質問という形でベクトル空間中の文書集合に問い合わせを行っている．検索質問は索引語の集合として表され，文書ベクトルと同様に検索質問ベクトルとして表すことができる．しかし，検索質問には多くの索引語を用いて検索することはほとんどないため，検索質問ベクトルの要素にはほとんど 0 となっている可能性が高い．ベクトル空間モデルにおける類似度計算は，検索質問に含まれる索引語で表現されたスパースな検索質問ベクトルから関連のある文書を見つけるために，比較が簡単な数値的な計算を用いて検索質問と検索対象の類似性を求める手法である．

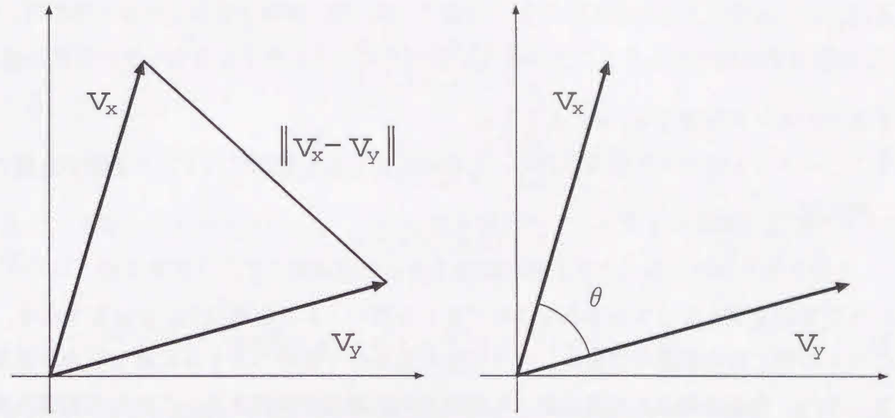

図 2.5 ふたつのベクトルの距離 (a) となす角 (b)

一般的に，観測されるベクトルは尺度を持つ空間として表さなければならない．尺度とは，要素のすべての対の間での距離や類似度などと呼ばれる $d(x,y)$ と表される関数によって定義された，どのようなベクトルに対しても持つ性質のことである．尺度に距離 (図 2.5(a)) を選んだ場合には，次のような条件が成り立っていなければならない．

$$\begin{cases} d(x,y) \geq 0, \text{ (等号は, } x = y \text{のときのみ成り立つ.)} \\ d(x,y) = d(y,x) \\ d(x,y) \leq d(x,z) + d(z,y) \end{cases} \tag{2.12}$$

ふたつのベクトルを $V_i = (w_{i1}, w_{i2}, \cdots, w_{in})$, $V_j = (w_{j1}, w_{j2}, \cdots, w_{jn})$ とする．このとき，V_i と V_j の距離 $d(V_i, V_j)$ を

$$d(V_i, V_j) = \|V_i - V_j\| \tag{2.13}$$

と定義する．ここで，距離としてよく用いられている直角座標系でのユークリッド距離は，

$$d(V_i, V_j) = \sqrt{(w_{i1} - w_{j1})^2 + (w_{i2} - w_{j2})^2 + \cdots + (w_{in} - w_{jn})^2} \tag{2.14}$$

と定義されている．

その他の距離の例としてはハミング距離があり，単純なものとして，2 進ベクトルに対して定義されている．2 進ベクトルは 0 または 1 のどちらかの要素しかとらないもので，ハミング距離は 2 つのベクトルの間でいくつの場所 (要素) が異なっているかを示している．

例えば，

$$x = (1,0,1,1,1,0)$$
$$y = (1,1,0,1,0,1) \tag{2.15}$$

とするとハミング距離 d は，$d = 4$ となる．

また，ベクトルがコード表現されているものに対しても同様にハミング距離が定義できる．例えば，

$$x = (p,a,t,t,e,r,n)$$
$$y = (w,e,s,t,e,r,n) \tag{2.16}$$

という要素が文字となっているふたつのベクトルのハミング距離 d は，$d = 3$ である．

これらの例にある距離の定義は，ベクトルの長さが一定のベクトルに対してのみ適用される．また，集合における表現の長さも制限される場合が存在する．これらの問題を解決する方法も多く存在し，単純な例としては，ベクトル V_i, V_j の要素の数を $n(V_i), n(V_j)$ として距離 d を次のように定義する．

$$d(V_i, V_j) = \max\{n(V_i), n(V_j)\} - n(V_i \cap V_j) \tag{2.17}$$

この距離尺度は順序付けされていない集合間で簡単で効果的な値が得られている [42].

この他にもコード化され，ベクトルが記号列になっているものの距離の定義に，レーベンシュタイン距離があり，次の式で与えられる．

$$LD(A,B) = \min\{a(i) + b(i) + c(i)\} \tag{2.18}$$

ここで，記号列 B は記号列 A での記号を $a(i)$ 回入れ換えて，$b(i)$ 回挿入し，$c(i)$ 回削除することにより得られるものである．しかし，この操作を行うとき $\{a(i), b(i), c(i)\}$ は無限の組合せが存在するので，動的計画法などでこの組合せの中から和が最小となるものを探し，それを距離とする．

これらの距離に対し，同様な類似性の尺度として類似度というものがある．文字通り，どれだけ類似しているかを数字によって表すことであるが，これは見方を換えれば，距離とは全く逆の考え方である．距離は長ければ長いほどふたつのベクトルの間の間隔が長くなり，近い位置には存在しないことになるため，非類似度と呼ぶこともできる．

検索質問と文書の類似度は 2 つのベクトルのなす角度を計算することにより，求めることができるが，ベクトル空間モデルでは一般的にベクトル間の余弦 (cosine) を用いて計算することが多い．余弦は 2 つのベクトルのなす角度を直接表しているので，値が 1 に近づくほど検索質問と文書はより関連性が強いことが分かり，値が 0 に近づくほど検索質問と文

書は関連性が弱いことが分かる．索引語・文書行列 A の各文書ベクトルを a_j $(j = 1, \cdots, d)$，検索質問ベクトルを q とすると，以下の余弦計算の式により d 個の類似度を得ることができる．

$$\cos\theta_j = \frac{a_j^T q}{\|a_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^m a_{ij} q_i}{\sqrt{\sum_{i=1}^m a_{ij}^2} \sqrt{\sum_{i=1}^m q_i^2}} \quad (j = 1, \cdots, d) \tag{2.19}$$

また，文書ベクトル a_j $(j = 1, \cdots, d)$ と検索質問ベクトルを q の距離が正規化されているときは，検索質問と文書の類似度は簡単な内積計算となり，以下の式で表される．

$$eval(a_j, q) = a_j^T q = \sum_{i=1}^m a_{ij} q_i \quad (j = 1, \cdots, d) \tag{2.20}$$

また，パターン認識の分野においてタニモトによって提案された尺度で，ふたつのベクトル間での類似度を以下に示す式によって定義すると，情報検索や分類，病名の判断などにおける類似性によい結果をもたらすとして，幾つかの実験により示されている．

$$S_T(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\|^2 + \|V_j\|^2 - V_i \cdot V_j} \tag{2.21}$$

この類似度の起源は集合の比較からなっている．例えば，A と B が文書の識別子や記述子，またはパターン内の離散的な特徴というような，はっきりとした数字ではない要素からなるふたつの集合であると考える．A と B の類似度は，その共通する要素の数とすべて異なった要素の数との比と定義してもよい．もし $n(X)$ が集合 X 中の要素の数とすると，類似度は以下のようになる．

$$S_T(A,B) = \frac{n(A \cap B)}{n(A \cup B)} = \frac{n(A \cap B)}{n(A) + n(B) - n(A \cap B)} \tag{2.22}$$

ここで，上で述べたふたつのベクトル V_i と V_j が特別な要素があるかないかによってその値が 0 または 1 の 2 進ベクトルであるとすると，$n(A \cap B)$，$n(A)$ と $n(B)$ は，$V_i \cdot V_j$，V_i と V_j に相当する．式 (2.22) を実数値ベクトルの範囲にまで拡張したものが式 (2.21) となる．

この他にもこれと同じような係数が多数存在しているが，様々なアルゴリズムに用いられている Dice 係数を以下に表す [33].

$$S(V_i, V_j) = \frac{2\sum_{k=1}^n (w_{ik} \cdot w_{jk})}{\sum_{k=1}^n w_{ik}^2 + \sum_{k=1}^n w_{jk}^2} \tag{2.23}$$

これらの類似度の尺度のうちで，どの尺度が最も良いかどうかの解説はなされていないので，一般的にはどのような尺度を用いても，それほど大きな変化は存在しないと考えられる．

2.9 Latent Semantic Indexing(LSI)

LSI は，ベクトル空間モデルの一種であり，特異値分解などのような行列の変換手法を用いて，多次元空間における文書ベクトルの要素を抽象化する．この LSI によく用いられる SVD は，一般に制約条件のない線形最小二乗問題の解や行列の階数や相関を求めるために使われる手法で，直交同値変換に基づいた行列の対角化が行われる．

2.9.1 特異値分解

n 個の文書からなる文書集合から，$m \times n$ である索引語・文書行列 A が得られたとする．このとき，行列 A の階数が r であるとすると，A の特異値分解 (Singular Value Decomposition, SVD) は次のように定義される．

$$A = U\Sigma V^T \tag{2.24}$$

ここで，$U = (u_1, \cdots, u_m)$ と $V = (v_1, \cdots, v_n)$ は $U^TU = V^TV = I_n$ を満たすユニタリ行列，Σ は $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_n)$，$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_n = 0$ を満たす対角行列で，この σ_i $(i = 1 \cdots r)$ は A の特異値と呼ばれる．これらの特異値の値 σ_i により，左特異ベクトル u_i と右特異ベクトル v_i が導き出され，A の i 番目の 3 つ組 $\{u_i, \sigma_i, v_i\}$ が定義される．この 3 つ組を用いることで，行列 A は次のように表すことができる．

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T \tag{2.25}$$

特異値分解によって求められる，行列 A の特異値の数は，この行列の階数 r に等しい．また，行列 A のフロベニウスノルム $\|A\|_F$ は，以下のように A の対角成分から直接求めることができる．

$$\|A\|_F = \|U\Sigma V^T\|_F = \|\Sigma V^T\|_F = \|\Sigma\|_F = \sqrt{\sum_{j=1}^{r} \sigma_j^2} \tag{2.26}$$

ここで，フロベニウスノルム $\|A\|_F$ は行列に対するノルムとして用いられるもので，以下の式で定義される．

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2} \tag{2.27}$$

行列 A との差のフロベニウスノルムが最小となる，階数 k 以下の行列 B を求めるとき，B は以下のように，特異値分解により階数を k に圧縮した行列になることが知られている．

$$\min_{rank(B) \leq k} \|A - B\|_F = \|A - A_k\|_F \tag{2.28}$$

元の 6×5 の索引語・文書行列 A を表す．

$$A = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1.0000 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$

特異値分解を用いて，階数を 3 として近似した行列 A_3 を示す．

$$A_3 = \begin{pmatrix} 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.6003 & 0.0094 & 0.9933 & 0.3858 & 0.7091 \\ 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \\ -0.0326 & 0.9866 & 0.0094 & 0.4402 & 0.7043 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \end{pmatrix}$$

図 2.6 索引語・文書行列から特異値分解により得られる階数 3 の近似行列 ([5] から引用)

ここで，A_k は $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ を満たす行列である．この式より，行列 A_k は，行列 A の階数を減少させた行列の中で A を最もよく近似した行列であることがわかる．また，近似した A_k と元の行列 A との誤差は，以下のようになる．

$$\|A - A_k\|_F = \min_{\text{rank}(X) \leq k} \|A - X\|_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2} \tag{2.29}$$

SVD は信号処理，パターン認識のための制御理論や実時間解析のなかでのさまざまな処理を行う重要な計算手法である．なかでも，Total Least Squares(TLS) は線形な等式を持つシステム $Ax \approx b$ を解くための便利な手法で，その最も一般的な TLS アルゴリズムは $[A|b]$ のような大きな行列の SVD を求めることに基づいている．この TLS は Frobenius ノルムにおける A と階数を下げた近似行列 B を最小にする行列を見つけるといい換えることができる．

$$\min_{B \in \mathcal{R}^{m \times n}, y \in \mathcal{R}^n} \|A - B\|_F, \text{ subject to } By = 0,\ y^Ty = 1 \tag{2.30}$$

これを Lagrange 乗数法を使って解くと，解は以下のような式になる．

$$Ay = x\tau,\ x^Tx = 1 \tag{2.31}$$

$$A^T x = y\tau,\ y^T y = 1 \tag{2.32}$$

このとき $\|A - B\|_F = \tau$ となり，最小の特異値を持つ 3 つ組 (x, τ, y) となる．

例として，図 2.6 に示した行列に対して特異値分解を行うと，分解されてできる 3 つの行列 $A = U\Sigma V^T$ は，以下のようになる．$Sigma$ は 4 個の 0 でない特異値を持ち，2 行の零ベクトルを持つ階数が 4 の行列で，U の初めの 4 行が，行列 A の列空間に対する基底となる．

$$U = \begin{pmatrix} 0.2670 & -0.2567 & 0.5308 & -0.2847 & 0.7071 & 0 \\ 0.7479 & -0.3981 & -0.5249 & 0.0816 & 0 & 0 \\ 0.2670 & -0.2567 & 0.5308 & -0.2847 & 0.7071 & 0 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 & 0 & 0.7071 \\ 0.5198 & 0.8423 & 0.0838 & -0.1158 & 0 & 0 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 & 0 & 0.7071 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1.6950 & 0 & 0 & 0 & 0 \\ 0 & 1.1158 & 0 & 0 & 0 \\ 0 & 0 & 0.8403 & 0 & 0 \\ 0 & 0 & 0 & 0.4195 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} 0.4366 & -0.4717 & 0.3688 & -0.6715 & 0 \\ 0.3067 & 0.7549 & 0.0998 & -0.2760 & -0.5000 \\ 0.4412 & -0.3568 & -0.6247 & 0.1945 & -0.5000 \\ 0.4909 & -0.0346 & 0.5711 & 0.6571 & 0 \\ 0.5288 & 0.2815 & -0.3712 & -0.0577 & 0.7071 \end{pmatrix}$$

ここで，最も特異値の小さい 4 番目の特異値を省略し，階数を 3 と近似した行列を A_3 とすると，元の行列 A との誤差は，$\|A - A_3\|_F = \sigma_4 = 0.4195$ となる．また，A のノルムは $\|A\|_F = 2.2361$ であるから，近似行列 A_3 が元の行列 A に対してノルムの損失する割合は，$|A - A_3\|_F / \|A\|_F \approx 0.1876$ となり，階数を 4 から 3 に下げた場合，ノルムは約 19% の損失を生じる．さらに，階数を 4 から 2 に下げると，ノルムが損失する割合は $|A - A_2\|_F / \|A\|_F \approx 0.4200$ となり，階数を 4 から 2 に下げた場合，ノルムは約 42% の損失を生じる．

2.9.2 次元圧縮時の類似度計算

検索質問ベクトル q と近似した索引語・文書行列 A_k との比較を行い，類似度の計算を行う．e_j を $n \times n$ である単位行列の j 行目のベクトルと定義すると，$A_k e_j$ は行列 A_k の j 行目のベクトルと表すことができる．このとき，検索質問ベクトル q と近似した文書ベクトルとの間のなす角の余弦は以下の式のようになる．

$$\cos\theta_j = \frac{(A_k e_j)^T q}{\|A_k e_j\|_2 \|q\|_2} = \frac{(U_k \Sigma_k V_k^T e_j)^T q}{\|U_k \Sigma_k V_k^T e_j\|_2 \|q\|_2} = \frac{e_j^T V_k \Sigma_k (U_k^T q)}{\|\Sigma_k V_k^T e_j\|_2 \|q\|_2} \tag{2.33}$$

ここで，$s_j = \Sigma_k V_k^T e_j$ と置き換えると，

$$\cos\theta_j = \frac{s_j^T (U_k^T q)}{\|s_j\|_2 \|q\|_2}, (j = 1, \cdots, n) \tag{2.34}$$

このように，余弦の計算は近似行列 A_k 明確にを求めることなしに計算をすることができる．また，ノルム $\|s_j\|$ は検索質問には関係なく，この値を一度だけ計算しておけば，すべての検索質問についての類似度計算に用いることができる．

ベクトル s_j の k 個の要素は，行列 U_k の列ベクトルの張る基底における，次元圧縮した行列 A_k の第 j 列に等しい．また，ベクトル $U_k^T q$ の k 個の要素は，検索質問ベクトル q を次元圧縮した行列 A_k に射影した行列 $U_k U_k^T q$ の基底に等しくなる．上の余弦計算の式を変更して，分母の検索質問ベクトル q を $U_k^T q$ に射影した式とすると，以下のようになる．

$$\cos\theta'_j = \frac{s_j^T (U_k^T q)}{\|s_j\|_2 \|U_k^T q\|_2}, (j = 1, \cdots, n) \tag{2.35}$$

この場合，余弦計算は検索質問ひとつにつき，$U_k^T q$ を一回計算するだけで済み，内積計算も k 次元空間内だけで計算をすることができる．検索質問ベクトル q は通常スパースなベクトルであるので，$U_k^T q$ を計算すること自体はそれほど計算量を必要とはしない．

図 2.6 を例として，"baking bread"についての文書を検索するための検索質問 q_1 を考えた場合，索引語・文書行列 A の階数を 3 とした行列 A_3 との比較を行うと，式 2.34 に示す余弦の値はそれぞれ 0.7327，−0.0469，0.0330，0.7161，−0.0097 となる．この結果から，1 番目と 4 番目の文書タイトルは高い類似度において正しく検索されているが，その他の文書については余弦がほとんど 0 に近いために，検索されていない．実際に，余弦がほとんど 0 に近い文書タイトルは与えられた検索質問とは，関連性が低いのが容易に分かるため，有効に検索が行われていると理解できる．

2.10 情報検索システムの評価

情報検索システムの性能を評価する目的は，システムの性能を向上させたり，複数存在するシステムのうち，どのシステムを使用するのが実際の検索に有効であるのかを比較，検討できるという目的がある．このようなシステムの性能を評価するためには，情報検索システムに関するさまざまな観点から考える必要がある．例えば，次に示すような観点がある．

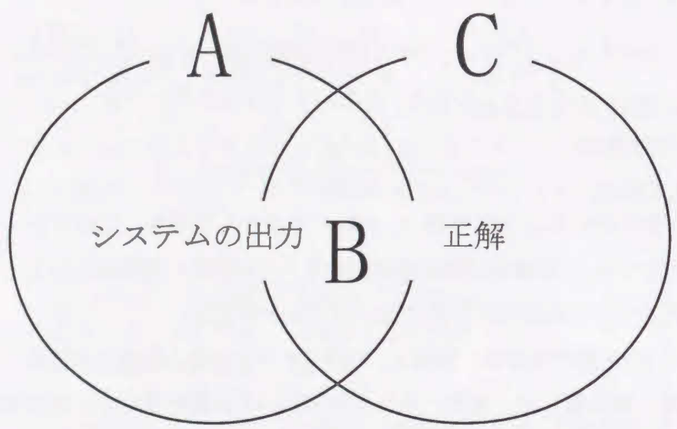

図 2.7 検索結果とテスト・コレクションとの適合文書集合の関係

(1) 有効性

情報検索システムの有効性とは，その検索システムを使うことによって，利用者が必要な情報を漏らすことなく，どれだけ正確に検索できるかによって決定される．この有効性は，再現率 (recall) や正解率 (precision) という指標で評価することが多い．有効性は，情報検索システム構築の際，基本技術の組合せをどのように選択するかによって，大きく影響を受ける．

(2) 効率性

システムの効率性とは，システムの応答時間，すなわち，利用者が検索質問を入力してからシステムが利用者に検索結果を返すまでにかかる時間がどの程度必要であるかという観点において評価を行う．これは，ファイルシステムの構造や索引語をデータベース化する際のデータ構造，あるいは記憶媒体など，ハードウェアの影響によるものが大きい．

本節では，このような観点を基に，検索結果を評価するための評価基準について概説する．個々のユーザの観点から見た場合，検索質問に対して適合する文書には統一性がほとんどないために，最終的な検索結果だけを用いて評価することは，現実的に不可能である．情報検索システムの評価では，個々のユーザが判定した場合に起こる関連文書の判定の異なり具合を考慮せず，検索質問に対して関連する文書の集合をあらかじめ定義しておき，これをもとに評価を行う．このように，文書集合と検索質問集合，さらに個々の検索質問に対する関連文書集合を定義した評価用データのことをテスト・コレクション (test collection)

と呼ぶ．テスト・コレクションに含まれる文書集合を用いて検索システムを構築し，個々の検索質問に対する検索システムの出力結果が，テスト・コレクションに用意されている関連文書集合にどの程度含まれているかによって評価を行う．

2.10.1 評価基準

情報検索における，システムの出力結果の評価方法には，一般的に再現率 (recall) と適合率 (precision) が用いられる．再現率は，検索質問に対するシステムの出力がどのくらい正解と関連するのかを示す指標で，実際の関連する文書数に対するシステムが出力した関連文書数の割合を示す．適合率は，実際に関連した文書をどれだけ検索できるかを表す指標で，システムが出力した検索結果中に存在する関連文書数の割合である．この定義から，適合率は検索能力を示し，再現率は検索の幅を示していると言い換えることができる．

再現率と適合率の定義を示すために，ある文書集合と検索質問について，それぞれの検索質問に対する各文書の適合性がテスト・コレクションによって与えられているとする．このとき，検索システムの出力した文書集合とテスト・コレクションで定義された関連のある文書集合の関係を示したものを図 2.7 に示す．ここで，図 2.7 の A は検索システムが出力した文書数，B は検索システムの出力した文書とテスト・コレクションでの正解文書が共通している文書数，Cは，テスト・コレクションでの正解の文書数を表す．このとき，再現率，適合率はそれぞれ式 (2.36)，式 (2.37) で定義される．

$$recall = \frac{B}{C} \tag{2.36}$$

$$precision = \frac{B}{A} \tag{2.37}$$

ベクトル空間モデルを用いた検索システムのように，検索結果をランキングとして出力するシステムを評価する場合には，一般的に再現率-適合率曲線が用いられる．これは，特定の再現率における点 (例えば，11 点:0.0 $\cdots$ 1.0 まで 0.1 きざみ) での適合率をグラフ化したものである．しかし，単にこのようなグラフ化を適用しただけでは，再現率 0.0 などといった低い再現率における適合率が確定できなくなる場合があるため，一般的に，補間再現率-適合率曲線が用いられる．これは，例えば，再現率 0.1 での適合率の値は，再現率が 0.1 以上の範囲における最大の適合率を取るという方法である．

例として，ある検索質問を別々の検索システムにおいて検索を行ったときの検索結果を，それぞれ表 2.4，表 2.5に示す．これらの検索結果からの再現率-適合率曲線を示したものを，

表 2.4 検索結果の例 1

文書番号	ランク	正否	再現率	適合率
45	1	○	0.20	1.00
23	2	×	0.20	0.50
89	3	×	0.20	0.33
98	4	○	0.40	0.50
44	5	○	0.60	0.60
90	6	×	0.60	0.50
7	7	×	0.60	0.43
9	8	×	0.60	0.38
51	9	○	0.80	0.44
31	10	○	1.00	0.50

表 2.5 検索結果の例 2

文書番号	ランク	正否	再現率	適合率
89	1	×	0.00	0.00
45	2	○	0.20	0.50
31	3	○	0.40	0.67
98	4	○	0.60	0.75
44	5	○	0.80	0.80
23	6	×	0.80	0.67
7	7	×	0.80	0.57
9	8	×	0.80	0.50
51	9	○	1.00	0.55
90	10	×	1.00	0.50

図 2.8に示す.

2.11 結言

以上のように，現在までに様々な情報検索の手法が考案されているが，それぞれについて長所と短所を兼ね備えている．検索検索システムの構築を行う場合，用いる検索手法に対して明らかになっている長所や短所の他に，これまでに明らかになっていない様々な要因によって，情報検索システムに対していくらかの影響を与えていると考えられる．このように，システムの性能向上を目指すためには，多くの要因についての議論が必要となり，日々活発に研究が行われている．

このような数多く存在する要因の中で改善することができる，何らかの理論を立てることから始め，そこから具体的な計算方法を導き，検索実験をテストコレクションを用いて行った結果を検証するという一連の流れが，情報検索技術研究の流れになっている．近年

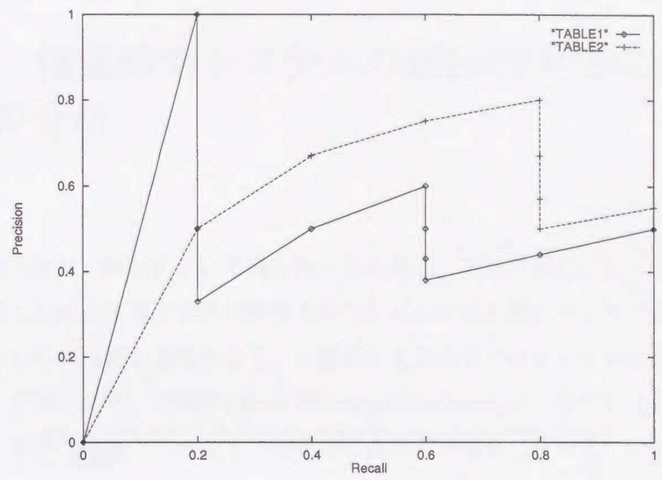

図 2.8 表 2.4，表 2.5の再現率-適合率曲線

では，多くの研究グループが共通のデータベース，プラットフォームにおける情報検索システムの実験結果を評価する場として，成果を相互比較するワークショップが盛んに行われ，研究推進の大きな原動力になっている．

次章では，このようなワークショップのひとつである IREX に参加したシステムの手法的な特徴をシステムに関するアンケート結果から抽出し，検索性能との相関関係を分析する．この分析結果より，ユーザの検索質問やシステム設計者の目的にあった情報検索システムを構築するための，有効な手法の選択についての客観的な評価を行う．

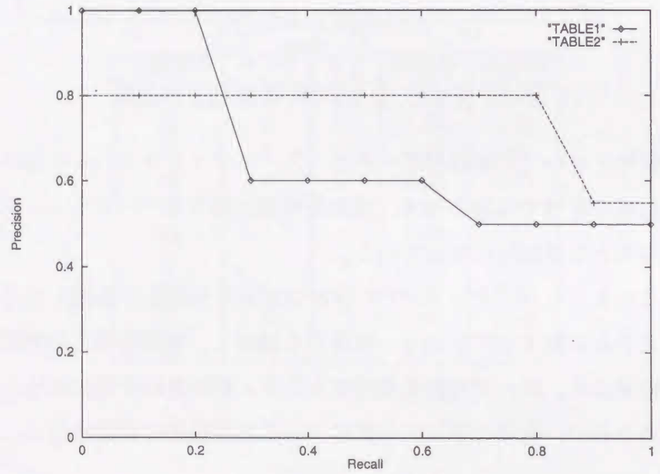

図 2.9 図 2.8の補間再現率-適合率曲線

第 3 章 情報検索システムの統計的手法による特徴と精度の分析

3.1 緒言

情報検索の分野は，欧米において過去数十年の間に，英語を中心とした文書を対象に研究が盛んに進められ，高速な文字列検索アルゴリズムや自動索引づけなどに多くの成果が得られた．これらの技術が基礎となり，大規模な文書集合に対する検索技術，新しい評価技術の向上を目的として，TREC (Text REtrieval Conference) [1] などのコンテストが開催され，新しい技術の開発やこれまでの技術の改良などが活発に行われている．

日本においても，情報抽出，検索技術に関する研究が盛んに行われ，数多くの優れた日本語情報検索システムが提案されている．このようなシステムを評価するための日本語テストコレクションの整備も進み [20]，個々の検索システムを容易に評価できるようになった．さらに，共通のデータベース，プラットフォームにおけるシステム評価の場として，IREX (Information Retrieval and Extraction Exercise) ワークショップ[2] が開催された．このワークショップには，情報検索 (IR) と情報抽出 (NE) の各課題に対して数多くのシステムが参加し，全体的な評価を通して様々な議論が行われた．

IREX の目標のひとつとして，共通の基準における各検索システムの評価を基にした問題点の共有と，それによるこの分野の飛躍的な進歩，発展がある [38]．一般的に，情報検索システムの性能評価をする際には，提案された手法を利用したシステムと利用していないシステムとの比較を行う．比較する際，一つのシステムからみると，参加した数多くのシステムにおける評価結果の違いから，研究の新しい方向性や発展性が発見できる．しかし，IREX では，数多くのシステムが参加しているため，ふたつのシステム間の比較実験では実験回数が莫大となり，共通点，相違点の整理が複雑になってしまう．また，他の比較手法として，使用されたシステムとは別の基準システムを作り，比較を行う手法も提案されている [17]．しかし，その場合，システム間の相違点が多くなり，直接的に何が精度向上の原因であるのかをとらえることが難しくなる．したがって，すべての検索システムを対象として，システムの構成要素を評価すると同時に，全体的なシステムの検索精度を評

[1] TREC ワークショップホームページ: http://trec.nist.gov
[2] IREX ワークショップホームページ: http://cs.nyu.edu/cs/projects/proteus/irex/

価するようなシステム指向の評価方法が必要となる．このような全体的な評価は，問題点を発見，解決するための議論を進める上で重要な課題であると考えられ，TREC や IREX では様々な評価が行われている [24][28][47].

本論文では，IREX における IR 課題の本試験の結果，および参加した各システムについての，参加者が回答したアンケート結果を参考にして，IR 課題におけるシステムの特徴と精度の関連性を独自の統計的な手法を用いて分析する．これまでは，手法を利用したシステムと利用していないシステムとの実験結果を比較することによって，その手法の有効性が評価されていた．これに対し，我々の提案する評価手法は，数多くのシステムにおける検索結果を基にして，システムに用いられた手法との関連性を客観的な相関係数として表し，検索システムに対し有効な手法を明確にしている．このような検索システムに対し有効な手法を示す評価は，これまで TREC7 においても行われているが，比較に用いられたすべてのシステムで再現率・適合率曲線の違いがほとんど無い条件の下で行われている [49]．この条件において，比較に用いたシステムが利用した手法が示されているが，客観的にその手法が有効かどうかの判断は難しい．その点で，我々の評価手法はどのような再現率・適合率曲線に対しても客観的に有効な手法を示すことができる．さらに，我々の評価手法は，検索結果でのランクの上位に，関連のある文書を数多く検索するための有効な手法を示すことができる．この分析においては，IR システムのアンケートの中でシステムの性能に大きく影響する次の 3 点

- 索引づけ，索引構造
- 検索式の生成
- 検索モデル，ランクづけ

に注目して，これらの要素を実現するために用いられた手法が検索精度とどの程度関連があるのかを調査する．

IREX では，図 3.1 に示す検索課題の例のように，検索要求を簡潔に表現した DESCRIPTION タグと，人間が判断可能な程度の詳細な検索要求の記述をした NARRATIVE タグが用いられている．通常，WWW サイトなどに存在する検索エンジンに入力される索引語の数は 2，3 語と少ないために，DESCRIPTION タグのみを検索実験に考慮する方が実用的である．しかし，DESCRIPTION タグのみを利用した場合には曖昧さが生じてしまい，人間が可能な限り正確に検索できるという点においては，詳細に書かれている NARRATIVE タグの方が重要な情報であるといえる．実際，TREC などにおいても，このような検索要

図 3.1 IREX ワークショップにおける検索課題の例

```
<TOPIC>
<TOPIC-ID>1014</TOPIC-ID>
<DESCRIPTION>金融機関の不良債権の処理</DESCRIPTION>
<NARRATIVE>銀行、証券などの金融機関における不良債権の具体的な処理についての
記事。金融機関や政府、日銀などが不良債権の処理に対して具体的な決定をした施策
や経営改善計画の発表などについて述べられた記事。<NEG>国民や評論家、当局担当者
個人の意見、まだ案の段階のもの、引用的に用いられているものなど、具体性に欠く
ものは除く。</NEG></NARRATIVE>
</TOPIC>
```

求の長さに対する精度への影響が議論され，NARRATIVE タグの使用による精度の違いが分析されている [18][48][49]．このようなことから，IREX においても，検索式を作成する際の NARRATIVE タグの使用有無により，検索システムに与える影響が変化するものと考えられる．このことを明らかにするため，検索システムにおける NARRATIVE タグの利用有無により short と long に分け，それぞれの平均適合率と相関の高いシステムの特徴を調べる．

再現率・適合率曲線に対し単回帰分析を行い直線として近似した場合，その切片が大きい時，ランクの上位に適合する文書を検索できる確率が高いと考えられる．また，傾きが平行に近いほど，システムは再現率の増加とともに起こる適合率の減少を抑えることができると考えられる．そこで，検索結果を平均して得られた再現率・適合率曲線に単回帰分析を行い直線として近似し，その切片と傾きがさまざまな手法のなかでどの手法に関連性が強いのかを調べる．また，同様に，short と long における切片と傾きとの相関が高いシステムの特徴を調べる．これらを分析することにより，本試験に参加したすべてのシステムで，検索質問を short と long に分けたそれぞれの場合に対して，傾き，切片から総合的に，どの手法と関連性が強いかを考察する．

3.2 再現率と適合率の関係

IR システムの評価には，一般的に適合率 (Precision) と再現率 (Recall) が使用される [26][50]．適合率は，システムが検索した文書に対する，検索した正解文書数の割合であり，検索の能力を表している．再現率は，全正解文書数に対するシステムが検索した正解文書

表 3.1 判定結果記事数 ([38] から引用)

課題番号	A 判定	B 判定	判定対象記事総数	課題番号	A 判定	B 判定	判定対象記事総数
予備試験				1018	55	101	2086
1001	80	145	931	1019	42	45	1859
1002	89	61	1096	1020	94	173	1291
1003	42	407	1316	1021	58	68	2030
1004	108	66	1480	1022	19	31	2015
1005	50	41	1099	1023	33	68	2853
1006	66	77	1356	1024	60	74	2934
本試験				1025	67	138	2047
1007	175	300	2246	1026	72	165	1914
1008	29	73	2565	1027	65	165	2513
1009	99	125	1588	1028	100	115	2806
1010	14	29	2222	1029	23	62	1878
1011	88	158	2130	1030	92	121	2053
1012	25	42	1535	1031	109	178	2134
1013	199	260	1308	1032	44	78	2268
1014	141	260	1473	1033	9	49	2989
1015	132	176	1505	1034	60	131	1911
1016	43	45	2446	1035	53	88	2008
1017	20	81	2248	1036	32	88	2299

数の割合であり，システムがすべての適合する文書のうちどの程度実際に検索可能かという検索の幅を表している．再現率と適合率はそれぞれ個別に用いてもシステムの評価を行うことができるが，ランクづけを行う検索システムでは，一般的に再現率・適合率曲線が用いられている．

IREX ワークショップにおける IR 課題は，2 年分の新聞記事から検索課題に書かれた検索要求に関連する文書を検索するもので，予備試験と本試験が行われた．表 3.1 に示すように，予備試験は課題数が 6 課題あり評価結果は非公開，本試験は課題数が 30 課題あり，評価結果は実際の団体名が分からないように各団体をシステム ID により表し，公開している．検索結果の判定基準には A，B，C の 3 種類あり，A 判定は記事の主題が検索課題に関連している場合，B 判定は A 判定のように記事の主題には関連性がないが，記事の一部が関連している場合，C 判定は何も関連していない場合という判断基準になっている．その IR の本試験に参加したすべてのシステムにおける，A 判定のみを正解とした検索結果を図 3.2 に示す．このグラフにおいて，一つの曲線を示す‘1103a’などの文字列はシステム

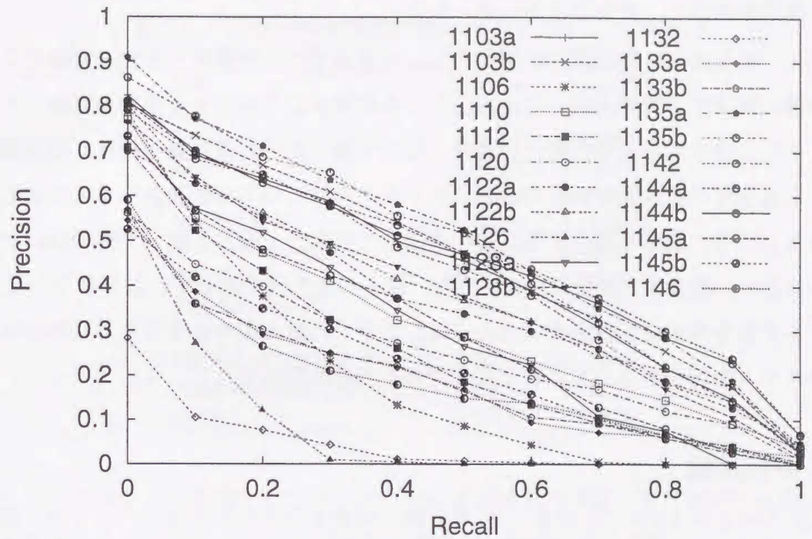

図 3.2 A 判定のみの再現率・適合率曲線

ID を表している．このグラフから分かるように，再現率と適合率の関係は大きく分けて 2 種類あると考えられる．一つ目は，適合率が再現率の増加に対し，直線的に減少する関係である．これは，ほとんどのシステムに当てはまる傾向で，特に再現率が 0.0 での適合率の値が高いシステムがこのようなグラフになっている．もう一つはグラフが下に凸の曲線を描くように適合率が減少する関係である．このような曲線は，検索の結果上位にランクされている文書に適合する文書が少ないため，再現率が少ない値において適合率の変動が激しくなっていると考えられる．システムについてのアンケートから，この曲線になる直接的な原因を調査したが，特にシステムに共通して用いられている手法は存在しなかった．

また，多くのシステムが再現率 0.0 における適合率が 0.7 を超えており，ランク上位に適合する文書が検索される確率が高くなっている．再現率 0.0 における適合率の値はランク上位に適合する文書を検索できるのかを表す尺度で，高い値をもつシステムほどユーザの探している情報が検索されていると考えることができる．ランクの 1 位で検索された文書が適合する文書であると，その時点で再現率 0.0 での適合率は 1.0 となる．適合しない文書の場合には適合率は 1.0 にはならず，以降の検索結果のランクに従って適合率に大きな変動が生じる．変動の大きさはあるものの，その後適合する文書が適合しない文書に比べ

数多く検索されると，適合率は高い値となる．

しかし，再現率 0.0 での適合率の大きさには関係なく，再現率・適合率曲線のグラフは必ず単調に減少する特徴を持っている．ランクに関係なく適合する文書が連続して検索される，または適合する文書の割合が適合しない文書に比べ大きい場合には，再現率の増加に対する適合率の減少量が少なくなる．ランクの上位にいくらか適合しない文書を検索していたとしても，途中で適合する文書を連続して検索し，適合しない文書の数に比べて多くなることで，適合率の減少が少なくなり，傾きの最大値である 0 に近くなる．これより，適合する文書を効率良く検索するためには，グラフの減少量を少なくする必要があり，これはシステムを評価する上で重要な尺度であると考えられる．

3.3 評価実験

IREX ワークショップにおける，IR 本試験の結果を公表する方法については，団体名を実名で公表するのではなく，各団体に割り当てられた ID 番号で結果を公表する方法がとられた．このため，検索結果に対してどのような手法が用いられているのか，研究的な内容を対応づけることが難しくなっている．システムの詳細を知る手段として，検索結果と同時に提出した IR システムアンケートがある (付録 A 参照)．このアンケートにより，各システムがどのような手法を利用したかが理解できるようになっている．

IR システムアンケートは，各システムについてどのような手法を用いて本試験の検索結果を出したのかを回答したものであり，主に次の項目がある．

- 索引づけとそのデータ構造
- 検索式の作成
- 検索を実行する環境
- 検索モデル
- その他

これらの項目の回答 (付録 B 参照) を集計して，システムが用いた手法の内，主要な 55 個の手法に注目した．これらの手法を平均適合率などの数値と比較するために，手法の使用の有無により数値を割り当てる．本実験では，各システムが用いた手法には 1 を，用いていない手法については 0 を割り当て，数値データに変換した．このときアンケートの中で「はい」,「いいえ」で答えられない質問項目については，システムにただひとつしか用いられていない手法でもひとつの手法として数値を割り当て，システムの違いを明確に定める

表 3.2 平均適合率と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
LSI	0.04545	-0.49600
IDF	0.86364	0.49341
レレバンスフィードバック	0.22727	0.45499
名詞	0.45455	-0.42376
フレーズ	0.31818	0.41665
NEG タグ	0.36364	0.40655
ロバートソン法	0.09091	0.40427
文書の長さ	0.45455	0.38297
名詞以外の品詞	0.40909	-0.38053
シソーラス	0.13636	0.33728
DESCRIPTOIN タグ	0.90909	-0.33728
文字列形態素インデクス	0.09091	0.32822
BM25	0.09091	0.32822

ことにした．このように変換をしたデータに平均適合率を付け足してひとつの行列にして，用いられた手法と平均適合率との相関係数を求めた．相関とは，変数 x，y において一方の変化が他方の変化にある傾向を伴うとき，x と y の間に相関関係があるという．相関には次の 3 種類がある．

正の相関：一方の値が大きくなると，他方の値も大きくなる．

負の相関：一方の値が小さくなると，他方の値も小さくなる．

無相関：2 つの値に明白な関係がみられない．

この相関を数値的に表したものに相関係数があり，2 つの変数の相互関係の程度を表したものである．本実験では，相関の有無を明確にするために，相関係数の絶対値が 0.5 を超える手法は相関が認められるとして，システムが用いた手法に対して評価を行った．

3.3.1 平均適合率とシステムの関連

平均適合率とシステムとの相関係数を求めた結果，相関係数の高かった主なシステムの特徴を表 3.2に示す．相関係数の高い手法は，全体的に正の相関を持っているが，すべての手法に対し，相関係数の絶対値が 0.5 を下回っている．表 3.2において，相関係数の絶対値の高い LSI(Latent Semantic Indexing) や IDF(Inverse Document Frequency) については，若干の相関は見られるものの，0.5 に満たしていないため，平均適合率との間に明確な相関を認めることができなかった．

また，この表 3.2を見ると，DESCRIPTOIN タグから索引語を抽出する手法と NEG タグなどのように NARRATIVE タグを利用した手法が混在していることが分かる．実際に，

表 3.3 short の平均適合率と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
IDF	0.88888	0.64416
LSI	0.11111	-0.64416
フレーズ	0.55555	0.61796
レレバンスフィードバック	0.33333	0.57326
確率と情報量	0.33333	0.56095
名詞	0.66667	-0.54080
名詞以外の品詞	0.66667	-0.54080
構文解析	0.22222	-0.43654
NEG タグ	0.22222	0.42834
シソーラス	0.22222	0.42834
単語	0.77778	-0.42834

NARRATIVE タグ自体を利用することについては，平均適合率との相関係数が −0.03864 と低く，NARRATIVE タグは他の 2 つのタグに比べ，適合率との関連性が少ない結果となった．これは，NARRATIVE タグの中には比較的長い文章が存在し，検索に重要な索引語が存在するのと同時に，一般的に広い分野で使われる索引語も比較的多く存在しているため，このタグを使う際には注意が必要であると考えられる．このようなことから，IREX ワークショップに参加したシステムを検索課題を簡潔に表現した DESCRIPTOIN タグのみを用いたシステムと比較的長い文章が存在し，NEG タグ[3]が存在する NARRATIVE タグを同時に用いたシステムに分けて評価を行う．これにより，相関の認められる手法がより顕著に現れ，より有効な評価をすることができると考えられる．

3.3.2 short および long での平均適合率とシステムの関連

DESCRIPTION タグのみを用い，NARRATIVE タグを利用しない場合を short，検索課題をすべて使用した場合を long として，それぞれを用いたシステムにおける平均適合率と利用した手法との関連性を比較する．そのために，short，long それぞれにおける平均適合率とシステムが用いた手法との相関係数を求め，NARRATIVE タグを利用しない場合にはどのような手法が有効であるか，また，NARRATIVE タグを利用した場合についても同様に有効な手法にどのようなものがあるかを調査する．その結果，short を用いたシステム，long を用いたシステムの平均適合率との相関係数が高かった主なシステムの特徴を，それぞれ表 3.3と表 3.4に示す．

表 3.3と表 3.4から共通して言えることは，short と long に分ける前の表 3.2と比較すると相関係数が全体的に高くなっているということである．すなわち，short と long に分けることによって，それぞれの検索システムに対して，平均適合率により関係深いシステムの特徴が顕著になっている．

[3]NARRATIVE タグ中において,「〜を除く」などの否定的な表現を示すもの

表 3.4 long の平均適合率と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
固有名詞	0.15385	0.74158
ロバートソン法	0.15385	0.74158
構文的な手がかり	0.15385	0.74158
文書の長さ	0.46154	0.68955
NEG タグ	0.46154	0.48372
名詞	0.30769	-0.41387
面の情報	0.15385	-0.41384
IDF	0.84615	0.39269
単語	0.61538	0.37213
形態素解析	0.15385	-0.36284

short で平均適合率と相関が最も高かったのは，LSI と IDF である．LSI は負の相関を持ち，検索精度を下げる傾向があるため，今回の本試験での結果においては，LSI の利用や索引づけに改良が必要であった．平均適合率と直接関連のある手法に，文書の重みづけとしては IDF，索引語には意味を限定しやすいフレーズが，共に正の相関を持ち検索精度を上げる傾向がある．フレーズを用いるとき，索引語が数多く存在しない場合があるため，レレバンスフィードバックを用いて検索式を拡張させることで，性能の良いシステムが構築できると考えられる．名詞や名詞以外という品詞情報を用いて索引づけを行う方法は，どちらも負の相関が高くなった．これは，元々索引語の数などのように情報量が少ない上に，一般的な単語まで取り出していたために，検索精度が下がる傾向があると考えられる．

long で平均適合率と相関が最も高かったのは，固有名詞である．これは，NARRATIVE タグにある文章が比較的長いめ，索引づけの手法の中でも特定分野にしか出現しない語を抽出できる固有名詞の相関が高くなったと考えられる．また，ロバートソン法や構文的な手がかり，文書の長さといった文書そのものの違いや特徴を明確にする重みづけの手法も相関が高かった．short では平均適合率に関連性があった IDF は，long の場合，これらの手法より相関が低かった．このことより，索引語に対する重みづけ手法は関連性が低いと考えられる．

表 3.3に見られる手法と表 3.2に見られる手法を比較すると，同じような手法が全体に見受けられた．すなわち，short の場合は，一般的な検索システムの平均適合率に関連性のある手法がそのまま関連性があるということができる．long の場合は，表 3.4に示した相関

表 3.5 回帰係数と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
文書の長さ	0.45455	-0.60396
構文解析	0.04545	-0.49385
IDF	0.86364	-0.44741
固有名詞 (索引)	0.22727	-0.43614
固有名詞 (検索式)	0.22727	-0.36891
単語	0.68182	-0.30771
面の情報	0.09091	0.30267
LSI	0.04545	0.30140

表 3.6 回帰直線の定数項と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
IDF	0.86364	0.60051
文書の長さ	0.45455	0.55917
LSI	0.04545	-0.55286
ロバートソン法	0.09091	0.44812
NEG タグ	0.36364	0.42826
フレーズ	0.31818	0.39632
固有名詞 (検索式)	0.22727	0.39059
レレバンスフィードバック	0.22727	0.37556
面の情報	0.09091	-0.36234
照合文字列長	0.33400	-0.34285

と全体の結果とを比較すると，全体的に正の相関を持つものが多かったが，表 3.2では見られなかった手法が多く現れた．

これらの結果から， NARRATIVE タグの使用有無によって検索システムの平均適合率に関連性のある手法が変化することが分かった．したがって，NARRATIVE タグを用いる際には，平均適合率に関連性のある，適した手法を選択する必要があると考えられる．

3.3.3 回帰式とシステムの関連

検索結果を平均して得られた，再現率と適合率の関係のデータを単回帰分析を用いて直線近似を行う．計算の結果，傾き (回帰係数) や切片 (定数項) に対して相関の高かった手法を，それぞれ表 3.5と表 3.6に示す．直線回帰を行ったときの決定係数は最大で 0.99446，最小で 0.69193 となり，22 個のシステムにおける平均が 0.942 と高い値となり，再現率・適合率曲線を直線で近似することが妥当であることを表している．

傾きと最も相関係数の高いものは文書の長さで，相関係数が -0.60396 と相関の認められる数値で，傾きを下げる傾向がある．これは，検索した文書数が増えるにつれて，適合する文書を検索する割合が少なくなることを表している．特に，傾き，切片共に関連が大

表 3.7 short の回帰係数と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
シソーラス	0.12500	0.75160
LSI	0.12500	0.75160
自動検索質問拡張	0.12500	0.75160
IDF	0.75000	-0.74788
フレーズ	0.37500	-0.57108
文字列形態素インデクス	0.25000	-0.51490
確率と情報量	0.25000	-0.51490
BM25	0.25000	-0.51490

表 3.8 short の回帰直線の定数項と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
IDF	0.75000	0.60051
シソーラス	0.12500	0.55917
LSI	0.12500	-0.55286
自動検索質問拡張	0.12500	0.44812
文字列形態素インデクス	0.25000	0.42826
確率と情報量	0.25000	0.39632
BM25	0.25000	0.39059

きい IDF と文書の長さは重みづけ手法であり，文書間での違いや特徴を明確にしているため，検索システムの構築において，重みづけ手法が特に重要であると考えることができる．

切片と最も相関係数の高いものは IDF で，次いで，文書の長さ，LSI となっている．全体的に正の相関を持つ手法が多く，切片を上げる傾向がある．しかし，これらの手法には，傾きにおいて負の相関の高い手法と共通するものが多く存在している．これより，これらの手法に対して，切片の大きさと傾きの大きさにはトレードオフの関係が存在していることが分かる．

3.3.4 short および long での回帰式とシステムの関連

NARRATIVE タグの有効性，有効な利用方法について考えるため，3.3.2節と同様に short と long に分割し，評価を行った．傾きや切片に対して相関の高かった手法を，short の場合をそれぞれ表 3.7，表 3.8に，long の場合をそれぞれ表 3.9，表 3.10に示す．

short における関連性

short の場合，傾きと最も相関係数の高い手法は，シソーラス，LSI，自動検索質問拡張で，これら 3 つは等しい値で，正の高い相関を持つ．しかし，全体的には負の相関を持つ手法が多く，傾きを下げる傾向がある．short では情報が少ないために，索引づけされた語をあらかじめ準備した知識集合であるシソーラスで拡張することにより，傾きを上げる高

表 3.9 long の回帰係数と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
ベクトル	0.14285	-0.60621
文書の長さ	0.14285	-0.60621
索引語の長さ	0.14285	0.59558
IDF	0.42857	-0.58837
フレーズ	0.28571	-0.47094
インデクス (ベクトル)	0.92857	0.38171
TF	0.07142	-0.38171
構文的な手がかり	0.07142	-0.38171

表 3.10 long の回帰直線の定数項と相関の高い主なシステムの特徴

システムの特徴	平均	相関係数
フレーズ	0.28571	0.75913
ベクトル	0.14285	0.62619
文書の長さ	0.14285	0.62619
索引語の長さ	0.14285	-0.60014
NEG タグ	0.57142	0.52856
IDF	0.42857	0.49117
名詞	0.28571	-0.46559
n-gram	0.28571	0.36211

い正の相関が得られたものと考えられる．さらに，LSI，フレーズ，自動検索質問拡張，文字列形態素インデクスもシソーラスと同様に索引づけの手法で，相関係数は比較的高い値になっている．このことから，検索に有効な索引語の選択や LSI による意味的な表現形式を得る手法が，傾きに深く関わっているといえる．

また，切片と最も相関係数の高い手法は IDF で，次いで，シソーラス，LSI となっている．こちらは，全体的に正の相関を持つものが多く，切片を上げる傾向がある．傾きとの相関の高い手法と共通するものが多く，切片でも索引づけの手法が深く関連しているといえる．シソーラスや自動検索質問拡張は傾き，切片ともに正の相関係数を持ち，short の場合，シソーラスが切片と傾きに最も関連の深い手法だと考えられる．しかし，LSI，IDF などは，傾きでは正の相関，切片では負の相関を持っているために，ここでも傾きと切片の間にトレードオフの関係が存在することが分かる．

long における関連性

long の場合，傾きと最も相関係数の高い手法は，ベクトル，文書の長さで，次いで索引語の長さとなる．short と同様に，全体的に負の相関を持つ手法が多く，ベクトルなどは傾きを下げる傾向がある．また，検索要求全体における傾き，切片と同様に，文書の長さ，

索引語の長さ，IDF などの重みづけ手法が深く関わっていることが分かる．さらに，ベクトルは文書または検索要求全体を用いた手法であるため，long を扱う際の索引語の増加に従って，関連性がより顕著に現れたと考えられる．

切片と最も相関係数の高い手法はフレーズで，次いで，ベクトルと文書の長さが高い相関を持っている．short 同様，正の相関を持つものが多く，切片を上げる傾向がみられるが，これらのほとんどは重みづけの手法である．しかし，これらの手法は，傾きと負の相関の高い手法と多数共通しているため，long においてもトレードオフの関係が存在していることが明らかになった．また，検索要求の長い long の場合，正の相関が高い NEG タグは，傾きとの相関係数が低く，切片との相関係数は高い．したがって，NEG タグの利用は，ランクの上位に適合する文書を検索する有効な手段であると考えることができる．

3.4 結言

本章では，IREX ワークショップにおける IR の本試験の結果，および，参加したすべての IR システムについてのアンケートを基に，平均適合率，再現率・適合率曲線を直線回帰させた傾きと切片が IR システムに用いられた手法とどのような相関関係をもっているのかを調査し，それぞれの手法がシステムの性能に与える影響の大きさを示した．その結果，名詞や名詞以外の品詞単語を用いる以上にフレーズが性能向上に関係あり，複数の単語を組み合わせることで意味が限定され，精度に良い影響を与えることを確認することができた．また，再現率・適合率曲線の切片と傾きにトレードオフの関係が多くの手法に見られ，検索システムに用いる手法の選択の難しさが現れる結果となった．

さらに，NARRATIVE タグの使用有無により short と long に分け，平均適合率との相関関係，また，再現率・適合率曲線を直線回帰させた傾きと切片にそれぞれどのような相関関係があるかを調査し，システムの性能に与える影響の大きさを示した．その結果，分ける前と比較して，全体的に相関が高くなり，関連が大きいシステムの特徴が顕著に現れるようになった．また，NARRATIVE タグを利用する場合，それに適した有効な手法を選択することが重要であることが分かった．short では，分ける前と比較して，索引語そのものに対する索引づけの手法への関連性がより顕著に現れた．その中でも，シソーラスや自動検索質問拡張は検索性能を上げる関連の深い手法であった．一方，long では文書全体に対する重みづけの手法が性能向上に関係があった．形態素解析やそれにより抽出される名詞単語を用いるよりも，フレーズや固有名詞などを用いる方が性能向上に関連が深く，より

要求する意味が明確になっている．このことは形態素解析により索引語を選択する難しさを表現し，形態素から索引語をより有効的に選択する手法が必要であることを示している．

第 4 章 ランダム・プロジェクションによる次元縮退を用いたベクトル空間情報検索モデル

4.1 緒言

近年，インターネットの普及とともに，個人で WWW (World Wide Web) を代表とするネットワーク上の大量の電子データやデータベースが取り扱えるようになり，膨大なテキストデータの中から必要な情報を取り出す機会が増加している．しかし，このようなデータの増加は必要な情報の抽出を困難とする原因となる．この状況を反映し，情報検索，情報フィルタリングや文書クラスタリング等の技術に関する研究開発が盛んに進められている．

情報検索システムの中でよく使われている検索モデルに，ベクトル空間モデル [37] がある．ベクトル空間モデルは，文書と検索要求を多次元空間ベクトルとして表現する方法である．基本的には，文書集合から索引語とするタームを取り出し，タームの頻度などの統計的な情報により，文書ベクトルを表現する．この際，タームに重みを加えることにより，文書全体に対するタームの特徴を目立たせることが可能である．この重みを計算するために，IDF(Inverse Document Freqency)[8] などの重みづけ方法が数多く提案されている．また，文書と検索要求を比較する類似度の尺度として，内積や余弦 (cosine) がよく用いられている．この類似度計算により，類似度の高いものからランクづけを行い，ユーザに表示することができることもベクトル空間モデルの特徴のひとつである．

ベクトル空間モデルを用いた検索システムを新聞記事などの大量の文書データに対して適用した場合，文書データ全体に存在するタームの数が非常に多くなるため，文書ベクトルは高い次元を持つようになる．しかし，ひとつの文書データに存在するタームの数は文書データ全体のターム数に比べると非常に少なく，文書ベクトルは要素に 0 の多い，スパースなベクトルになる．このような文書ベクトルを用いて類似度を計算する際には，検索時間の増加や文書ベクトルを保存するために必要なメモリの量が大きな問題となる．このため，単語の意味や共起関係などの情報を用いたり，ベクトル空間の構造を利用してベクトルの次元を圧縮する研究が盛んに行われている．このようなベクトルの次元圧縮技術には，統計的なパターン認識技術や線形代数を用いた手法などが用いられている [13][23]．この中で，最も代表的な手法として，LSI (Latent Semantic Indexing) がある [9][12]．この手法は，文書・単語行列を特異値分解を用いて，低いランクの近似的な行列を求めるものであり，こ

れを用いた検索システムは，次元圧縮を行わない検索モデルと比較して一般的に良い性能を示す．しかし，特異値分解に必要な計算量が大きいために，検索モデルを構築する時間が非常に長いことが問題となっている．

上記の問題を解決するベクトル空間モデルの次元圧縮手法に，ランダム・プロジェクション [3] が存在する．ランダム・プロジェクションは，あらかじめ指定した数のベクトルとの内積を計算することで次元圧縮を行う手法である．これまでに報告されているランダム・プロジェクションを用いた研究には，VLSI(Very Large-Scale Integrated circuit) の設計問題への利用 [45] や次元圧縮後の行列の特性を理論的に述べたものがある [3][31]．しかし，これらの文献では，ランダム・プロジェクションの理論的な特性は示されているものの，情報検索における具体的な実験結果は報告されていない．そのため，情報検索に対するランダム・プロジェクションの有効性に疑問が残る．

我々は，ランダム・プロジェクションを用いた情報検索モデルを構築し，評価用テストコレクションである MEDLINE を利用した検索実験を行った．この検索実験より，情報検索における次元圧縮手法として，ランダム・プロジェクションが有効であることを示す．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルとして，文書の内容を表す概念ベクトル [10] の利用を提案する．概念ベクトルは文書の内容が似ているベクトル集合の重心で，この概念ベクトルを得る際，高次元でスパースな文書データ集合を高速にクラスタリングすることができる球面 k 平均アルゴリズム [10] を用いる．これにより，文書集合を自動的にクラスタリングできるだけでなく，ランダム・プロジェクションに必要な概念ベクトルも同時に得ることができる．この概念ベクトルをランダム・プロジェクションで用いることにより，任意のベクトルを用いた検索性能と比較して，検索性能が改善されていることを示し，概念ベクトルを利用した次元圧縮の有効性を示す．

4.2 ランダム・プロジェクションによるベクトルの次元圧縮

本節では，ランダム・プロジェクションを用いたベクトル空間モデル [3][31] についての概観を述べる．ランダム・プロジェクションは，ひとつの文書データを n 次元空間上のベクトル $\mathbf{u}$ として表現するとき，このベクトルを k $(k < n)$ 次元空間に射影する手法である．その際，k 個の任意の n 次元ベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を用意する．用意したこれらのベクトルと n 次元ベクトル $\mathbf{u}$ の内積，

$$\mathbf{u}'_1 = \mathbf{r}_1 \cdot \mathbf{u}, \cdots, \mathbf{u}'_k = \mathbf{r}_k \cdot \mathbf{u} \tag{4.1}$$

をそれぞれ計算する．その結果，k 次元に圧縮した $\mathbf{u}'_1, \cdots, \mathbf{u}'_k$ を要素とするベクトルが得られる．

次元圧縮に必要なベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を列ベクトルとする $n \times k$ の行列 $\mathbf{R}$ を用いると，求める k 次元ベクトルは

$$\mathbf{u}' = \mathbf{R}^T \mathbf{u} \tag{4.2}$$

となり，ランダム・プロジェクションは行列計算のみの簡単な形で表現することができる．この行列 $\mathbf{R}$ が任意の正規直交行列のとき，すなわち，行列 $\mathbf{R}$ の列ベクトルがすべて単位ベクトルで，かつ，相異なる列ベクトルが互いに直交していれば，ランダム・プロジェクションは射影前後におけるベクトル間距離を近似的に保存する特性を持っている．

4.3 概念ベクトルを用いたランダム・プロジェクション

ランダム・プロジェクションに必要な行列 $\mathbf{R}$ は，これまでの研究では正規分布などの確率分布をなす任意の行列が用いられている [3][6][14][21][31][45]．このような行列を用いて任意の部分空間に射影する場合，次元圧縮を行う前後の任意のベクトル間距離は近似的に保存されることが示されている [15][19]．しかし，任意の正規直交行列を用いる場合，次元圧縮を行う前後のベクトル間距離を保存する効果は得られたとしても，LSI のように，ベクトルの要素が抽象的な意味を持つ索引語の生成や内容的に関連のある文書をまとめる効果があるとは考えられない．このことから，LSI のような，情報検索に有効な索引語を生成するために，ランダム・プロジェクションの改良が課題となる．

このような課題を解決するものとして，ランダム・プロジェクションでベクトルを次元圧縮をした後，さらに特異値分解を行うことにより，LSI の効果を得る手法が提案されている [31]．この手法は，関連文書をまとめる効果を得ると同時に，特異値分解のみを用いた場合に比べ，モデル作成に必要な時間を短縮したものである．しかし，ランダム・プロジェクションと特異値分解は，共にベクトル間距離を保存する効果を持つ手法であるため，特異値分解が内容的に関連のある文書，あるいはタームをまとめるために適用されているとしても，これらの手法を同時に利用することは，検索モデルを構築する時間に関して，効率の良い手法であるとはいえない．さらに，非常に大きい次元数をもつ行列について考えた場合，特異値分解に多くの計算量が必要であることも問題となる．したがって，特異値分解により誤差を最小とする近似行列を得る代わりに，誤差は最小ではないものの，ランダム・プロジェクションのみを用いて LSI の効果を得ることで，より高速に検索モデルが

構築できるのではないかと考えられる．

これを実現するために，我々は，ランダム・プロジェクションにおける行列 $\mathbf{R}$ に，文書の内容を表現した概念ベクトルを利用することを提案する．概念ベクトルは，文書ベクトル集合をクラスタリングしてできたクラスタの，各クラスタに属する文書ベクトルの重心を正規化したベクトルとして表される．この概念ベクトルによる次元圧縮は，単にベクトルを近似するだけではなく，クラスタに属するベクトル集合の重心を求めることにより，ターム間で特徴づけられる隠れた関連性やタームの同義性と多義性を捉えることができる．クラスタリングにより得られた各クラスタは互いに異なる概念を持ち，これより得られる概念ベクトルが圧縮した空間の軸となるように用いられる．これにより，次元圧縮された行列は文書と概念ベクトルの類似度を表し，元の空間において内容の近い文書は，圧縮した空間においても近くなる可能性がある．また，類似しているが，異なるタームを使った文書の場合，元の空間では近くないが，圧縮した空間では近くなる可能性があり，検索性能が改善されると考えられる．さらに，多義語により元の空間において近いとされる文書どうしが圧縮した空間では遠くに離れ，誤った検索が取り除かれる可能性も期待できる．このように，これまで単語などが要素であったベクトルが，文書の内容を要素とするようなベクトルに変換され，文書を低い次元で，より検索性能が向上するベクトル表現ができると考えられる．

概念ベクトルからなる行列 $\mathbf{R}$ を求めるために，球面 k 平均アルゴリズム [10] と呼ばれるクラスタリング手法を用いる．球面 k 平均アルゴリズムは，目的関数が局所的に最大となるまで，高い次元でスパースな文書データ集合をクラスタリングする手法である．球面 k 平均アルゴリズムでは，ユークリッド空間内でベクトル間のなす角の余弦を類似度とし，多次元空間の単位円を分割することによりクラスタリングを行う．これにより，文書ベクトルの集合は指定した数の部分集合に分割され，各クラスタの中心を計算することで，容易に概念ベクトルを作ることができる．さらに，このアルゴリズムは文書ベクトルのスパースさを逆に利用して高速に収束する利点を持ち，得られる概念ベクトルは特異値分解を用いたものに非常に近いことが示されている [10]．

しかし，球面 k 平均アルゴリズムにより得られる概念ベクトルは一般的に直交性を満たしているとは限らないため，概念ベクトルをランダム・プロジェクションに適用するには疑問が生じる．先に述べたように，距離を保存するには正規直交性を満たすベクトルを利用する必要があるが，この概念ベクトルをランダム・プロジェクションに適用する場合，直

交性を満たしていないとしても独立であれば，任意の行列においても十分に距離を保存する可能性のあることが示されている [3]．球面 k 平均アルゴリズムでは，内容的に似通ったベクトルをクラスタとしてまとめるため，原理的には独立した概念ベクトルを生成すると考えられる．このため，直交性に関して，概念ベクトルをランダム・プロジェクションに適用するのは問題ないと考えられる．

本節では，まず，球面 k 平均アルゴリズムの概要を述べる前に，クラスタリングにより得られる概念ベクトルについて述べる．

4.3.1 概念ベクトル

ベクトルの集合をベクトル空間にプロットしたとき，同質のベクトルが多く存在する場合を除いて，いくつかのグループに分かれる．このようなグループはクラスタと呼ばれ，類似した内容をもつベクトルの集合が形成される．概念ベクトルはクラスタに属するベクトルの重心を求めることにより得られ，そのクラスタの内容を表す代表ベクトルである．

概念ベクトルを求める例として，正規化された N 個のベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を，異なる s $(s < N)$ 個のクラスタ $\pi_1, \pi_2, \cdots, \pi_s$ にクラスタリングすることを考える．このとき，ひとつのクラスタ π_j に含まれるベクトル $\mathbf{x}_i$ の平均である重心 $\mathbf{m}_j$ は以下のように表される．

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \tag{4.3}$$

ここで n_j はクラスタ π_j に含まれるベクトルの数を表す．ベクトルの重心は単位長にはなっていないので，そのベクトルの長さで割ることにより概念ベクトル $\mathbf{c}_j$ を得る．

$$\mathbf{c}_j = \frac{\mathbf{m}_j}{\|\mathbf{m}_j\|} \tag{4.4}$$

4.4 目的関数

k 平均アルゴリズムでは，目的関数は一般的に概念ベクトルとクラスタに属するベクトルとの距離の和

$$\sum_{\mathbf{x}_i \in \pi_j} \|\mathbf{m}_j - \mathbf{x}_i\| \tag{4.5}$$

を最小にするような概念ベクトルを求める，最小二乗法が用いられる．球面 k 平均アルゴリズムでは，このような最小化問題ではなく，ミクロ経済学の分野における，生産計画の最適化問題で扱われている目的関数を用いている [21]．これは，各クラスタ $\pi_j (1 \leq j \leq s)$

の密度を

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{4.6}$$

とし，クラスタの結合密度の和を目的関数としている．

$$D = \sum_{j=1}^{s} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{4.7}$$

クラスタの密度は，以下のコーシー・シュワルツの不等式より，任意の単位ベクトル $\mathbf{z}$ に対して，クラスタ π_j に含まれるベクトル x_i と概念ベクトルとの内積の総和が最大となる．

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{z} \leq \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{4.8}$$

また，クラスタの密度は，それに属するベクトル和の距離に等しくなるという特徴を持っている．

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j = \| \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \| \tag{4.9}$$

4.4.1 球面 k 平均アルゴリズム

4.4節で示した目的関数 D を最大にするように，ベクトルの集合を反復法によりクラスタリングする．文書ベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を s 個のクラスタ $\pi_1^\star, \pi_2^\star, \cdots, \pi_s^\star$ に分割するためのアルゴリズムを以下に示す．

1) すべての文書ベクトルを s 個のクラスタに任意に分割する．これらの部分集合を $\{\pi_j^{(0)}\}_{j=1}^s$ とし，これより求められた概念ベクトルの初期集合を $\{\mathbf{c}_j^{(0)}\}_{j=1}^s$ とする．また，t を繰り返しの回数とし，初期値は $t = 0$ である．

2) 各文書ベクトル $\mathbf{x}_i (1 \leq i \leq N)$ に対し，余弦が最も大きい，最も文書ベクトルに近い概念ベクトルを見つける．このとき，すべての概念ベクトルは正規化されているので，余弦は文書ベクトル $\mathbf{x}_i$ と 概念ベクトル $\mathbf{c}_j^{(t)}$ の内積を求めることと同値である．これにより，前回の繰り返しで求めた概念ベクトル $\{\mathbf{c}_j^{(t)}\}_{j=1}^s$ から，文書ベクトルが新たな部分集合 $\{\pi_j^{(t+1)}\}_{j=1}^s$ に分割される．

$$\pi_j^{(t+1)} = \{\mathbf{x}_i : \mathbf{x}_i^T \mathbf{c}_j^{(t)} \geq \mathbf{x}_i^T \mathbf{c}_l^{(t)}\} \ (1 \leq l \leq N,\ 1 \leq j \leq s) \tag{4.10}$$

ここで，$\pi_j^{(t+1)}$ は概念ベクトル $\mathbf{c}_j^{(t)}$ に近いすべての文書ベクトルの集合とする．

3) 新たに導かれた概念ベクトルの長さを正規化する．

$$\mathbf{c}_j^{(t+1)} = \frac{\mathbf{m}_j^{(t+1)}}{||\mathbf{m}_j^{(t+1)}||}, \quad (1 \leq j \leq s) \tag{4.11}$$

ここで，$\mathbf{m}_j^{(t+1)}$ はクラスタ $\pi_j^{(t+1)}$ の文書ベクトルの重心を表す．

4) 目的関数 $D^{(t+1)}$ の値を求め．前回の繰り返しにおける目的関数の値 $D^{(t)}$ との差を計算する．このとき，

$$\|D^{(t)} - D^{(t+1)}\| \leq 1 \tag{4.12}$$

を満たす場合，$\pi_j^\star = \pi_j^{(t+1)}$，$\mathbf{c}_j^\star = \mathbf{c}_j^{(t+1)}$ $(1 \leq j \leq s)$ とし，アルゴリズムを終了する．停止基準を超えていない場合は，t に 1 を加え，ステップ 2 に戻る．ここで，停止基準における目的関数の差は，文書数が約 4000 で，クラスタの数が 8 よりも大きい場合，収束した時の目的関数は 1000 を超えることがこれまでの研究で報告されている [10]．このため，繰り返しでの 1 以下の差は無視できるとし，便宜的に 1 という値を設定した．

4.5 実験

本節では，ランダム・プロジェクションを用いた検索モデルを構築し，その評価として，MEDLINE を用いた検索実験について述べる．

4.5.1 データ

実験で用いたデータは，情報検索システムの評価用テストコレクションである MEDLINE を利用した．MEDLINE は医学・生物学分野における英文の文献情報データベースで，検索の対象となる文書の件数は 1033 件で，約 1Mbyte の容量を持つテキストデータである．また，MEDLINE には 30 個の評価用検索要求文と各要求文に対する正解文書が用意されている．

MEDLINE に含まれている 1033 件の文書全体から，前処理として，“a” や “about” などの一般的な 439 個の英単語を不要語リストに指定して，文書の内容と関係のほとんどない単語は削除した．この後，接辞処理を行い，残った英単語を語幹に変換する処理を行った．この前処理の結果，文書全体に 5526 個あった単語から，4329 個の単語が索引語として抽出され，実験データとして用いた．

4.5.2 検索実験方法

実験では，MEDLINE から前処理により得られた索引語を要素とする文書ベクトルと検索要求ベクトルを作成し，比較することで検索スコアを計算する．文書ベクトルを作成するとき，ベクトルの要素には局所的，大域的な索引語の分布を考慮するために，索引語の頻度に重み付けした数値が用いられる．数多く提案されている重みづけ手法で，今回の実験では以下の式で定義された対数エントロピー重み [8] を用いた．L_{ij} は j 番目の文書に対

する i 番目の索引語への重み，G_i は文書全体に対する i 番目の索引語への重みを表す．

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \tag{4.13}$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \tag{4.14}$$

ここで，n は全文書数，f_{ij} は j 番目の文書に出現する i 番目の索引語の頻度，F_i は文書集合全体における i 番目の索引語の頻度を表す．これより，j 番目の文書から得られる文書ベクトルの i 番目の要素 d_{ij} は，

$$d_{ij} = L_{ij} \times G_i \tag{4.15}$$

となる．

得られた文書ベクトルから，球面 k 平均アルゴリズムを用い，これらの文書ベクトルより指定された数の概念ベクトルを作成する．作成した概念ベクトルを結合した行列に対し，ランダム・プロジェクションを行い，文書ベクトル，検索要求ベクトルの次元を削減する．次元の削減されたベクトルを用いて，内積の計算を行い，その値を各文書に対する検索スコアとする．これらのスコアのうち，上位 50 文書を検索結果として出力する．

検索システムの評価には，一般的に用いられている正解率 (Precision) と再現率 (Recall) を用いた [26][50].

$$\text{Recall} = \frac{\text{システムが出力した正解文書数}}{\text{全正解文書数}} \tag{4.16}$$

$$\text{Precision} = \frac{\text{システムが出力した正解文書数}}{\text{システムが出力した文書数}} \tag{4.17}$$

再現率と正解率は，それぞれ個別に用いて，システム評価を行うことができるが，本実験では，一般にランクづけ検索システムの評価に用いられる再現率・正解率曲線を用い，システムの評価を行った．この曲線は，各質問に対しひとつの曲線が作成されるが，本稿の検索システム評価には，全 30 個の質問に対する各再現率での平均を計算した再現率・正解率曲線を用いた．

4.6 実験結果および考察

4.6.1 次元数による比較

本実験では，ランダム・プロジェクションにより，ベクトルの次元を 100 から 900 まで圧縮した検索モデルについて，検索実験を行った．その結果，各次元における平均正解率

表 4.1 各次元数における平均正解率

次元数	ランダム・プロジェクション	平均正解率
100	あり	0.3982
200	あり	0.4711
300	あり	0.5154
400	あり	0.5231
500	あり	0.5673
600	あり	0.5748
700	あり	0.5822
800	あり	0.5979
900	あり	0.6037
1033	なし	0.4936

は表 4.1のようになった．平均正解率は，ベクトルの次元が大きくなるにつれて増加し，次元数 300 において，次元圧縮を行わないベクトル空間モデルよりも良い結果となった．また，次元数が 400 から 500 に変化させたときの平均正解率の増加が最も大きく，それ以降は変化の割合が少なくなっている．次元数を大きくすれば，検索に必要な計算量が増加する．このことから，効果的な検索を行うためには，全文書数の約半分に次元圧縮を行う必要があることが分かった．

4.6.2 検索モデル作成時間

検索モデルを作成する時間，および，一つの検索要求に対し，検索を行うために必要な時間を測定した結果を述べる．検索実験には，Ultra Sparc(330MHz) のマシンを使用し，ベクトルの次元を 500 とした結果，表 4.2に示すように，ランダム・プロジェクションを用いた場合，モデルを作成する時間は約 11 分必要であった．LSI の場合，SVD の計算については SVDPACK の中で最も高速な Lanczos 法を利用し，同様にベクトルの次元を 500 とした結果，モデルを作成する時間は約 24 分で必要であった．この結果，ランダム・プロジェクションは LSI に比べ，高速に検索モデルを構築することができた．

このモデル作成時間においては，メモリサイズの大きさによる，SVD の計算時間に与える影響が考えられる．スワップ領域を用いるほどの大規模なデータについては大きな影響を及ぼし，モデル作成の時間を多く必要とするが，本実験において用いたマシンには 640M バイトのメモリを搭載しているため，MEDLINE コレクションのような規模のデータに対しては，メモリサイズの影響はほとんどないと考えられる．

本実験で用いた MEDLINE には収録されているデータは 1033 件と比較的少ない．この

表 4.2 モデル作成時間とひとつの検索要求に対する検索時間

手法	モデル作成時間	検索時間
ランダム・プロジェクション	約 2 分	4 秒
LSI	約 24 分	4 秒

表 4.3 文書数の変化によるモデル作成時間

データ	ランダム・プロジェクション	LSI
MEDLINE	約 2 分	約 24 分
MEDLINE+CISI	約 14 分	約 26 分
MEDLINE+CISI+CRANFIELD	約 34 分	約 43 分

ため，文書数を変化させたときの検索モデル構築時間の変化について比較を行った．文書数を増加させるために，MEDLINE と同様なテストコレクションである CISI を併せた 2493 記事，さらに CRANFIELD を併せた 3893 記事について，それぞれの検索モデル作成時間を測定した．その結果，ランダム・プロジェクションと LSI のモデル作成時間は表 4.3のようになった．これより，文書数が増加に対して球面 k 平均アルゴリズムの 1 回の反復による計算量が大きくなるのであるが，ランダム・プロジェクションが検索時間に関しても有効であることが分かる．しかし，非常に大規模な文書数に対しては，より 1 回の反復による計算量が増加するため，反復計算を必要とせずに，球面 k 平均アルゴリズム並の概念ベクトルを得ることが課題となった．

4.6.3 他の検索モデルとの比較

ランダム・プロジェクションを用いた検索モデルに対して，モデルとしての有効性について評価をする．この評価をするために，次元圧縮をしていない元のベクトル空間モデルと特異値分解を用いた LSI による検索モデルについての検索実験も同時に行い，性能を比較した．このとき，比較として用いた LSI は，次元数 100 として次元圧縮した検索モデルを用いている．これらの検索モデルについて，同様に検索実験を行い，すべての検索質問の平均を求めた再現率・正解率曲線を図 4.1に示す．図 4.1において，横軸は再現率を表し，縦軸は正解率を表す．またグラフの ‘LSI100’ は次元数 100 の LSI，‘VSM’ は次元圧縮なしのベクトル空間モデル，‘RP500’，‘RP700’，‘RP900’ はランダム・プロジェクションによるそれぞれに示された次元数に圧縮したモデルの実験結果である．

その結果，ベクトル空間モデルと比較して，ランダム・プロジェクションを用いた検索モデルは，大幅に性能が改善されていることが分かった．また，次元数 100 の LSI と比較

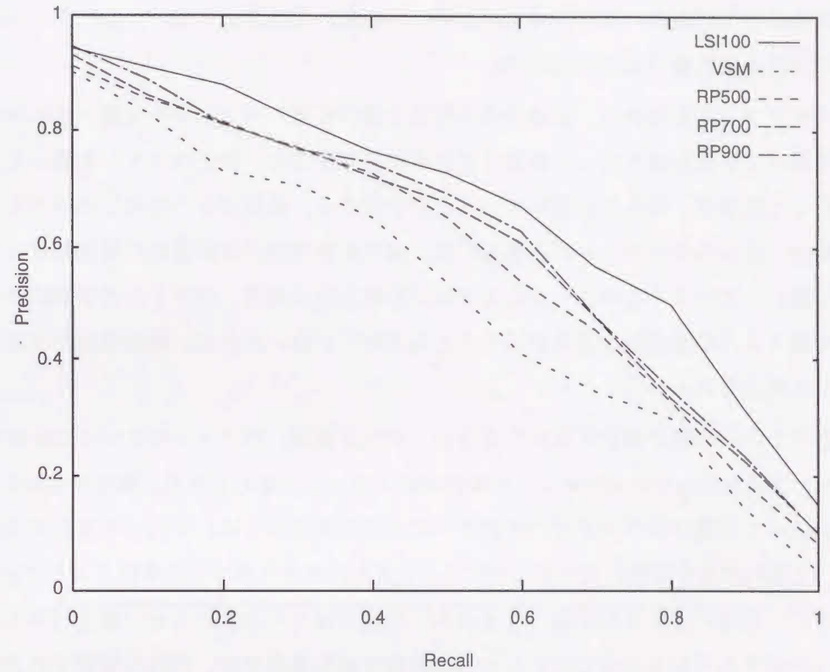

図 4.1 モデルに対する再現率・正解率曲線

すると，ランダム・プロジェクションは LSI に比べ少し下がってはいるものの，ほぼ同じ程度に検索精度が改善されていることを示している．このことから，ランダム・プロジェクションが検索モデルとして，LSI と同等の性能を持っていることが分かる．

4.6.4 概念ベクトルの有効性

ランダム・プロジェクションで次元圧縮に用いられる概念ベクトルが有効であるかを評価するために，他のベクトルを用いて次元圧縮が行われた場合との検索結果の比較を行った．ベクトルには，乱数を用いて，全要素の平均が 0，分散が 1 の正規分布 $N(0,1)$ となるベクトルと，指定された数の文書ベクトルを任意に抽出して得られた部分集合からなるベクトルを，それぞれ次元圧縮に用いた．この結果，再現率・正解率曲線は図 4.2となった．ここで，‘Random’ は正規分布となるベクトル，‘Subset’ は文書ベクトルの部分集合を表し，共にベクトルの次元数は 500 として，次元圧縮を行ったモデルの実験結果である．また，サンプルに使った文書集合の偏りを考慮するため，グラフに示した実験でのベクトルの他にいくらかのサンプルを用意し，同様の実験を行い，平均的な検索精度を求めた．その結果，

正規分布による任意のベクトルにおける平均正解率の平均値は 0.38，文書ベクトルの部分集合における平均値は 0.47 となった．

このグラフと平均値から，正規分布の性質を持つ任意のベクトルや文書ベクトルの部分集合を用いて次元圧縮を行った結果とそれぞれ比較すると，概念ベクトルを用いて次元圧縮を行った結果が，明らかに優れていることが分かる．乱数により生成したベクトルを用いた場合，これらのベクトルの各要素には，索引の重要度や索引語間の関連性はほとんど存在しない．このようなベクトルにより次元圧縮を行う場合，ベクトルの要素には文書の内容を表すような潜在的な意味がほとんど含まれていないために，検索性能が下がってしまったと考えられる．

文書ベクトルの部分集合を用いた場合は，次元圧縮後，ベクトル中のいくつかの要素が似通った意味を持っているために，検索性能が下がったと考えられる．概念ベクトルは，内容の似通った文書がクラスタリングによりひとつにまとめられ，それらの重心を求めることで，文書の内容を端的に表すことができる．また，クラスタリングを行うことで似通った内容を持つ概念ベクトルが少なくなるため，内容がほとんど変わらない概念ベクトルを重複して生成する可能性が少ない．しかし，文書の部分集合では，内容の重複した文書が複数存在する可能性がある．このため，次元圧縮後のベクトル空間モデルに意味の重なった要素が存在し，検索性能が下がってしまう可能性が大きくなってしまうと考えられる．これらのことにより，情報検索に対してランダム・プロジェクションを用いて次元圧縮を行う場合，内容の近い文書や同義語などのような索引語の特徴を表した概念ベクトルを用いることにより，優れた検索性能が得られることが示された．

4.7 結言

本論文では，ベクトル空間モデルの次元圧縮手法として，ランダム・プロジェクションを用いた検索モデルを提案した．このモデルの有効性を評価するために，MEDLINE を利用した検索実験を行った．その結果，次元圧縮していない元のベクトル空間モデルと比べ検索精度が改善されていることが分かった．また，LSI と比較しても，検索精度の差は少なく，ランダム・プロジェクションが LSI と同程度の次元圧縮性能を持っていることが分かった．LSI とランダム・プロジェクションのモデル作成，検索に必要な時間を比較すると，LSI は特異値分解を行うこともあり，ランダム・プロジェクションは LSI に比べ約半分の時間で検索を行うことができた．また，MEDLINE よりも大規模な文書集合に対して

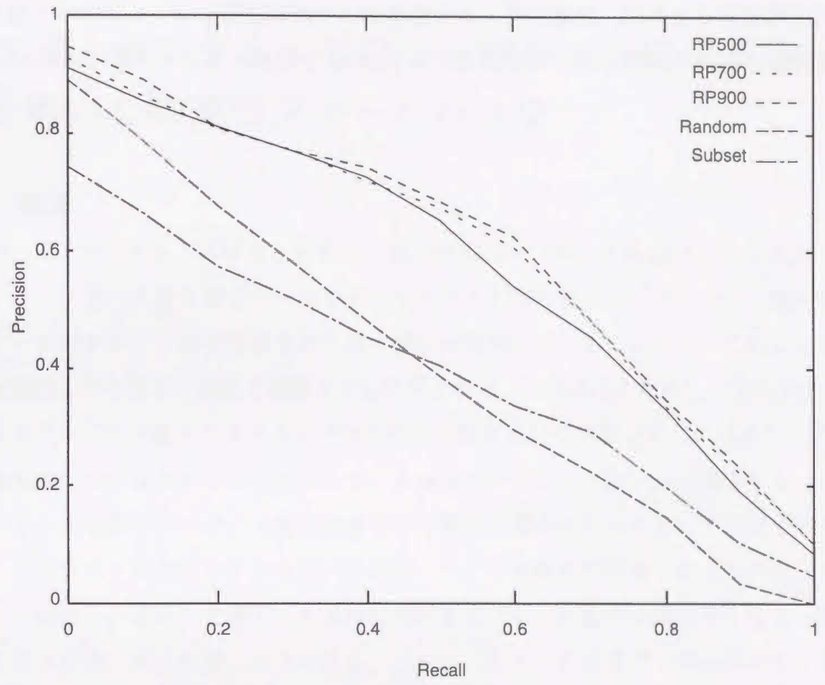

図 4.2 概念ベクトルに対する再現率・正解率曲線

も，ランダム・プロジェクションが高速に検索モデルが構築することができる．これらのことから，ランダム・プロジェクションは LSI に比べ，高速，かつ有効な次元圧縮手法であることが分かった．

また，ランダム・プロジェクションで次元圧縮に必要な行列を得るために，球面 k 平均アルゴリズムで得られる概念ベクトルの利用を提案し，その有効性を検索実験にて評価した．その結果，乱数により生成したベクトルや文書ベクトルの部分集合を用いた場合に比べ，検索精度が優れていた．文書間の内容などの特徴を表した概念ベクトルを用いることで，その概念における索引語の分布を，ベクトルのひとつの要素として表現することができる．これより，ランダム・プロジェクションを用いて検索モデルを構築するとき，概念ベクトルが潜在的な意味を有効にとらえることができることが分かった．

今後の研究課題としては，まず，球面 k 平均アルゴリズムは初期段階での分割に非常に大きな影響を及ぼす可能性があるため，初期分割に依存しない有効な概念ベクトルの生成方法を考慮し，より有効な次元圧縮を実現が可能であると考えられる．さらに，より有効

な次元圧縮を行うために，評価用データの解答やユーザの評価をフィードバック情報として，概念ベクトルの調節を行った検索モデル [40][46] を構築することが挙げられる．

第 5 章　ランダム・プロジェクションによる次元縮退を用いた関連性フィードバック

5.1　緒言

近年，インターネットの普及とともに，個人で WWW (World Wide Web) を代表とするネットワーク上の大量の電子データやデータベースが取り扱えるようになり，膨大なテキストデータの中から必要な情報を取り出す機会が増加している．しかし，このようなデータの増加は必要な情報の抽出を困難とする原因となる．この状況を反映し，情報検索，情報フィルタリングや文書クラスタリング等の技術に関する研究開発が盛んに進められている．

情報検索システムの中でよく使われている検索モデルに，ベクトル空間モデル [37] がある．ベクトル空間モデルは，文書と検索要求を多次元空間ベクトルとして表現する方法である．このベクトル空間モデルを用いた検索システムを新聞記事などの大量の文書データに対して適用した場合，文書データ全体に存在するタームの数が非常に多くなるため，文書ベクトルは高い次元を持つようになる．しかし，ひとつの文書データに存在するタームの数は文書データ全体のターム数に比べると非常に少なく，文書ベクトルは要素に 0 の多い，スパースなベクトルになる．このような文書ベクトルを用いて類似度を計算する際には，検索時間の増加や文書ベクトルを保存するために必要なメモリの量が大きな問題となる．このため，単語の意味や共起関係などの情報を用いたり，ベクトル空間の構造を利用してベクトルの次元を圧縮する研究が盛んに行われている．

上記の問題を解決するベクトル空間モデルの次元圧縮手法に，我々が提案したコンセプト・プロジェクションが存在する．コンセプト・プロジェクションは，クラスタリングなどにより得られる，文書の内容を表した概念ベクトルと文書ベクトルの内積を計算することで，次元圧縮を行う手法である．これにより，文書ベクトルは用意した概念ベクトルの数に次元圧縮され，検索時間が短縮されている．また検索性能に関しても，次元圧縮を行わないベクトル空間モデルよりも改善され，同様な次元圧縮手法である LSI(Latent Semantic Indexing) に匹敵する検索性能が得られている．

本稿では，我々の提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法について述べる．関連性フィードバックは検索結果の各文書が正解であるか，不正解であるかをユーザに判定させ，この判定評価の情報を用

いて初期検索要求に反映させる手法である．これに対し，提案するフィードバック手法は，判定評価の情報を初期検索要求に反映させるのではなく，コンセプト・プロジェクションの概念ベクトルに反映させている．これにより，更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われるため，フィードバック学習の影響が検索要求だけでなく検索対象にも反映させることができる．関連性フィードバックによる様々な概念ベクトルの更新手法を提案し，テストコレクションによる検索実験結果を示し，更新手法の比較を行う．

5.2　コンセプト・プロジェクションによるベクトルの次元圧縮

本節では，コンセプト・プロジェクションを用いたベクトル空間モデルの概観について述べる．まず初めに，コンセプト・プロジェクションに必要であるクラスタリングによって得られる概念ベクトルについて述べる．

5.2.1　概念ベクトル

ベクトルの集合をベクトル空間にプロットしたとき，同質のベクトルが多く存在する場合を除いて，いくつかのグループに分かれる．このようなグループはクラスタと呼ばれ，類似した内容をもつベクトルの集合が形成される．概念ベクトルはこのようなクラスタに属するベクトルの重心を求めることにより得られ，そのクラスタの内容を表す代表ベクトルである．

概念ベクトルを求める例として，正規化された N 個のベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を，異なる s $(s < N)$ 個のクラスタ $\pi_1, \pi_2, \cdots, \pi_s$ にクラスタリングすることを考える．このとき，ひとつのクラスタ π_j に含まれるベクトル $\mathbf{x}_i$ の平均である重心 $\mathbf{m}_j$ は以下のように表される．

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i \tag{5.1}$$

ここで n_j はクラスタ π_j に含まれるベクトルの数を表す．ベクトルの重心は単位長にはなっていないので，そのベクトルの長さで割ることにより概念ベクトル $\mathbf{c}_j$ を得る．

$$\mathbf{c}_j = \frac{\mathbf{m}_j}{\|\mathbf{m}_j\|} \tag{5.2}$$

5.2.2　コンセプト・プロジェクション

コンセプト・プロジェクションは，ひとつの文書データを n 次元空間上のベクトル $\mathbf{u}$ として表現するとき，このベクトルを k $(k < n)$ 次元空間に射影する手法である．その際，

クラスタリングなどにより求められた n 次元である k 個の概念ベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を用意し，これらのベクトルと n 次元ベクトル $\mathbf{u}$ の内積，

$$\mathbf{u}'_1 = \mathbf{r}_1 \cdot \mathbf{u}, \cdots, \mathbf{u}'_k = \mathbf{r}_k \cdot \mathbf{u} \tag{5.3}$$

をそれぞれ計算する．その結果，k 次元に圧縮した $\mathbf{u}'_1, \cdots, \mathbf{u}'_k$ を要素とするベクトルが得られる．

次元圧縮に必要なベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ を列ベクトルとする $n \times k$ の行列 $\mathbf{R}$ を用いると，求める k 次元ベクトルは

$$\mathbf{u}' = \mathbf{R}^T \mathbf{u} \tag{5.4}$$

となり，コンセプト・プロジェクションは行列計算のみの簡単な形で表現することができる．この行列 $\mathbf{R}$ が任意の正規直交行列のとき，すなわち，行列 $\mathbf{R}$ の列ベクトルがすべて単位ベクトルで，かつ，相異なる列ベクトルが互いに直交していれば，コンセプト・プロジェクションは射影前後におけるベクトル間距離を近似的に保存する特性を持っている．

概念ベクトルからなる行列 $\mathbf{R}$ を求めるために，球面 k 平均アルゴリズム [10] と呼ばれるクラスタリング手法を用いる．球面 k 平均アルゴリズムは，各クラスタ $\pi_j (1 \leq j \leq s)$ の密度を

$$\sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{5.5}$$

とし，クラスタの結合密度の和を目的関数とし，この目的関数が局所的に最大となるまで，高い次元でスパースな文書データ集合がクラスタリングされる．

$$D = \sum_{j=1}^{s} \sum_{\mathbf{x}_i \in \pi_j} \mathbf{x}_i^T \mathbf{c}_j \tag{5.6}$$

球面 k 平均アルゴリズムでは，ユークリッド空間内でベクトル間のなす角の余弦を類似度とし，多次元空間の単位円を分割することによりクラスタリングを行う．これにより，文書ベクトルの集合は指定した数の部分集合に分割され，各クラスタの中心を計算することで，容易に概念ベクトルを作ることができる．さらに，このアルゴリズムは文書ベクトルのスパースさを逆に利用して高速に収束する利点を持ち，得られる概念ベクトルは特異値分解を用いたものに非常に近いことが示されている [10]．しかし，球面 k 平均アルゴリズムにより得られる概念ベクトルは一般的に直交性を満たしているとは限らないため，概念ベクトルをランダム・プロジェクションに適用するには疑問が生じる．先に述べたように，距離を保存するには正規直交性を満たすベクトルを利用する必要があるが，この概念ベク

トルをランダム・プロジェクションに適用する場合，直交性を満たしていないとしても独立であれば，任意の行列においても十分に距離を保存する可能性のあることが示されている[3]．球面 k 平均アルゴリズムでは，内容的に似通ったベクトルをクラスタとしてまとめるため，原理的には独立した概念ベクトルを生成すると考えられる．このため，直交性に関して，概念ベクトルをランダム・プロジェクションに適用するのは問題ないと考えられる．

5.2.3　球面 k 平均アルゴリズム

5.2.2節で示した目的関数 D を最大にするように，ベクトルの集合を反復法によりクラスタリングする．文書ベクトル $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ を s 個のクラスタ $\pi_1^*, \pi_2^*, \cdots, \pi_s^*$ に分割するためのアルゴリズムを以下に示す．

1) すべての文書ベクトルを s 個のクラスタに任意に分割する．これらの部分集合を $\{\pi_j^{(0)}\}_{j=1}^s$ とし，これより求められた概念ベクトルの初期集合を $\{\mathbf{c}_j^{(0)}\}_{j=1}^s$ とする．また，t を繰り返しの回数とし，初期値は $t=0$ である．

2) 各文書ベクトル $\mathbf{x}_i (1 \leq i \leq N)$ に対し，余弦が最も大きい，最も文書ベクトルに近い概念ベクトルを見つける．このとき，すべての概念ベクトルは正規化されているので，余弦は文書ベクトル $\mathbf{x}_i$ と 概念ベクトル $\mathbf{c}_j^{(t)}$ の内積を求めることと同値である．これにより，前回の繰り返しで求めた概念ベクトル $\{\mathbf{c}_j^{(t)}\}_{j=1}^s$ から，文書ベクトルが新たな部分集合 $\{\pi_j^{(t+1)}\}_{j=1}^s$ に分割される．

$$\pi_j^{(t+1)} = \{\mathbf{x}_i : \mathbf{x}_i^T \mathbf{c}_j^{(t)} \geq \mathbf{x}_i^T \mathbf{c}_l^{(t)}\} \quad (1 \leq l \leq N,\ 1 \leq j \leq s) \tag{5.7}$$

ここで，$\pi_j^{(t+1)}$ は概念ベクトル $\mathbf{c}_j^{(t)}$ に近いすべての文書ベクトルの集合とする．

3) 新たに導かれた概念ベクトルの長さを正規化する．

$$\mathbf{c}_j^{(t+1)} = \frac{\mathbf{m}_j^{(t+1)}}{||\mathbf{m}_j^{(t+1)}||}, \quad (1 \leq j \leq s) \tag{5.8}$$

ここで，$\mathbf{m}_j^{(t+1)}$ はクラスタ $\pi_j^{(t+1)}$ の文書ベクトルの重心を表す．

4) 目的関数 $D^{(t+1)}$ の値を求め．前回の繰り返しにおける目的関数の値 $D^{(t)}$ との差を計算する．このとき，

$$\|D^{(t)} - D^{(t+1)}\| \leq 1 \tag{5.9}$$

を満たす場合，$\pi_j^* = \pi_j^{(t+1)}$，$\mathbf{c}_j^* = \mathbf{c}_j^{(t+1)}$ $(1 \leq j \leq s)$ とし，アルゴリズムを終了する．停止基準を超えていない場合は，t に 1 を加え，ステップ 2 に戻る．ここで，停止基準における目的関数の差は，文書数が約 4000 で，クラスタの数が 8 よりも大きい場合，収束した時の目的関数は 1000 を超えることがこれまでの研究で報告されている[10]．このため，繰り返しでの 1 以下の差は無視できるとし，便宜的に 1 という値を設定した．

5.3　フィードバックによる概念ベクトルの更新手法

情報検索システムはユーザに検索結果を提示し，ユーザはその結果に対して関連のある文書であると判定する．適合性フィードバックはその判定結果を元に，システムの挙動を変化させるようにパラメータを調節し，システムに反映させるものである．この関連性フィードバックがパラメータを調節する対象としては，検索質問，検索対象となる文書，または検索モデルが考えられる．よく知られているフィードバックに検索質問拡張があるが，検索質問のみを修正することは，システムに対して長期的な効果が得られるとは限らない[46][41]．本節では，テストコレクションに用意されている，検索質問に対してどの文書が適合しているかという情報を用いて，コンセプト・プロジェクションにおける概念ベクトルのパラメータ更新手法を提案し，検索精度の改善を試みる．

概念ベクトルのパラメータ更新の基本は，上位に検索された関連のある，または関連のない文書ベクトルをそれぞれ k 個の概念ベクトル $\mathbf{r}_1, \cdots, \mathbf{r}_k$ に加えて更新をする．これにより，概念ベクトルの持つ文書の内容がより検索質問の内容に近づき，検索精度が向上することが期待できる．また，検索結果からコンセプト・プロジェクションの概念ベクトルを，より検索したい内容の概念ベクトルに変更し，次元圧縮後は検索要求だけでなく，検索対象にもフィードバックを行うことができる．

具体的に，文書ベクトルを概念ベクトルに加える手法として以下の 5 種類を考慮し，関連のある場合，ない場合に対してこれらの手法を組み合わせて実験を行う．

1) 文書ベクトルとの内積が最も大きい概念ベクトル $\mathbf{r}_l$ を見つけ，その概念ベクトルに文書ベクトルを加えて，正規化を行う．

2) 文書ベクトルとの内積がある閾値 σ 以上の概念ベクトル集合を見つけ，それらの概念ベクトルにそれぞれ文書ベクトルを加えて，正規化を行う．

3) システムが検索した関連のある，または関連のない文書集合の重心との内積が，最も大きい概念ベクトル $\mathbf{r}_l$ を見つけ，その概念ベクトルに文書ベクトルを加えて，正規化を行う．

4) システムが検索した関連のある，または関連のない文書集合の重心との内積が，ある閾値 γ 以上である概念ベクトル集合を見つけ，それらの概念ベクトルにそれぞれ文書ベクトルを加えて，正規化を行う．

5) 適合性フィードバックの基本である手法で，i 回目の検索質問ベクトル Q_i から $i+1$ 回目の検索に向けて索引語の重みを修正した検索質問ベクトル Q_{i+1} を求める式を以下のように表した手法である [34].

$$Q_{i+1} = Q_i + \alpha \sum_{\mathbf{x} \in R_r} \mathbf{x} - \beta \sum_{\mathbf{y} \in R_n} \mathbf{y} \tag{5.10}$$

ここで，R_r は i 回目において検索された関連文書集合，R_n は i 回目において検索された関連のない文書集合である．また，α，β は定数であり，それぞれ関連文書，関連のない文書をどの程度重要視するかを調整する．

6) フィードバックは先の 5に示した手法で行い，検索質問ベクトルを修正した後で，コンセプト・プロジェクションを行い次元圧縮を行う．

5.4　実験

5.4.1　実験の概要と結果

コンセプト・プロジェクションを用いたフィードバック検索モデル構築し，その検索性能を示す実験を行った．実験には，情報検索システムの評価用テストコレクションである MEDLINE を利用した．MEDLINE は医学・生物学分野における英文の文献情報データベースで，検索の対象となる文書の件数は 1033 件で，約 1Mbyte の容量を持つテキストデータである．また，MEDLINE には 30 個の評価用検索質問文とそれらの関連記事が用意されている．

まず，前処理として MEDLINE の記事全体から抽出した 1033 件の記事から一般的な 439 個の英単語をストップワードに指定して，文書の内容と関係のほとんどない単語は削除した．この前処理の結果，4329 個のタームが索引語として抽出された．

これらの索引語を要素とする文書ベクトルを作成するとき，索引語の頻度に重みを加えた数値をベクトルの要素とする．数多く提案されている重みづけ手法で，今回の実験では以下の式で定義された対数エントロピー重み [8] を用いた．L_{ij} は j 番目の文書に対する i 番目のタームへの重み，G_i は文書全体に対する i 番目のタームへの重みを表す．

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \tag{5.11}$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i} \log \frac{f_{ij}}{F_i}}{\log n} \tag{5.12}$$

ここで，n は全文書数，f_{ij} は j 番目の文書に出現する i 番目のタームの頻度，F_i は文書集合全体における i 番目のタームの頻度を表す．

得られた文書ベクトルから，球面 k 平均アルゴリズムを用い，これらの文書ベクトルより指定した 500 の概念ベクトル作成する．作成した概念ベクトルを結合した行列に対し，ランダム・プロジェクションを行い，文書ベクトル，検索質問ベクトルの次元を 500 に削減する．次元の削減されたベクトルに対し，内積の計算を行い，その値を各文書ベクトルへの検索スコアとする．これらのスコアのうち，上位 50 文書を検索結果として，出力する．

検索システムの精度の評価には，一般的に用いられている適合率 (Precision) と再現率 (Recall) を用いた [26][50].

$$\text{Recall} = \frac{\text{システムが出力した適合文書数}}{\text{全適合文書数}} \tag{5.13}$$

$$\text{Precision} = \frac{\text{システムが出力した適合文書数}}{\text{システムが出力した文書数}} \tag{5.14}$$

再現率と適合率は，それぞれ個別に用いて，システム評価を行うことができるが，本実験では，一般にランクづけ検索システムの評価に用いられる再現率-適合率曲線を用い，システムの評価を行った．本稿の検索システム評価には，繰り返しの回数に従って，平均適合率がどのように変化しているのか示し，その中でもっともフィードバックの効果の高かった手法についての、全質問に対する各再現率での平均を計算した再現率-適合率曲線を示すことにより行った．この概要の元で，先に示した手法を用いて様々な実験を行った結果，フィードバックの有効な効果が得られた手法の，繰り返しの回数による平均適合率の変化を表 5.1 から 5.8に表す．

5.4.2　考察

これらの表からも分かる通り，コンセプト・プロジェクションを用いて 5 回のフィードバックを行った結果，最小で約 0.08，最大では約 0.21 の平均適合率の上昇が見られた．これにより，コンセプト・プロジェクションによるフィードバック手法の有効性を示すことができた．また，図 5.1では，フィードバックによる繰り返し回数に従って再現率-適合率曲線の適合率の減少率が少なくっていることも分かる．これは，フィードバックの結果，関連のある文書がより上位に検索されるためで，このグラフからもこの手法の有効性が理解できる．これらのことは，フィードバックが行われることにより，これまで文書集合をク

表 5.1 各繰り返し回数での平均適合率 1

	適合:手法 3, 不適合:手法 4($\gamma = 0.3$)
繰り返し回数	平均適合率
1	0.5552
2	0.6715
3	0.6723
4	0.6782
5	0.6861

表 5.2 各繰り返し回数での平均適合率 2

	適合:手法 4($\gamma = 0.3$), 不適合:なし
繰り返し回数	平均適合率
1	0.4893
2	0.6903
3	0.7017
4	0.6635
5	0.6892

表 5.3 各繰り返し回数での平均適合率 3

	適合:手法 4($\gamma = 0.3$), 不適合:手法 2($\sigma = 0.5$)
繰り返し回数	平均適合率
1	0.5216
2	0.6885
3	0.7303
4	0.6975
5	0.6807

表 5.4 各繰り返し回数での平均適合率 4

	適合:手法 4($\gamma = 0.3$), 不適合:手法 1
繰り返し回数	平均適合率
1	0.5118
2	0.6536
3	0.7314
4	0.7079
5	0.7289

表 5.5 各繰り返し回数での平均適合率 5

	適合:手法 4($\gamma = 0.4$), 不適合:手法 2($\sigma = 0.5$)
繰り返し回数	平均適合率
1	0.5434
2	0.6714
3	0.7140
4	0.7419
5	0.7532

表 5.6 各繰り返し回数での平均適合率 6

	手法 5($\alpha = 1.0$, $\beta = 0.5$)
繰り返し回数	平均適合率
1	0.4936
2	0.8662
3	0.9361
4	0.9593
5	0.9587

表 5.7 各繰り返し回数での平均適合率 7

	手法 6($\alpha = 1.0$, $\beta = 0.5$)
繰り返し回数	平均適合率
1	0.5682
2	0.5687
3	0.6178
4	0.6411
5	0.6451

表 5.8 各繰り返し回数での平均適合率 8

	手法 6($\alpha = 1.0$, $\beta = 0.0$)
繰り返し回数	平均適合率
1	0.5682
2	0.6599
3	0.6613
4	0.6623
5	0.6629

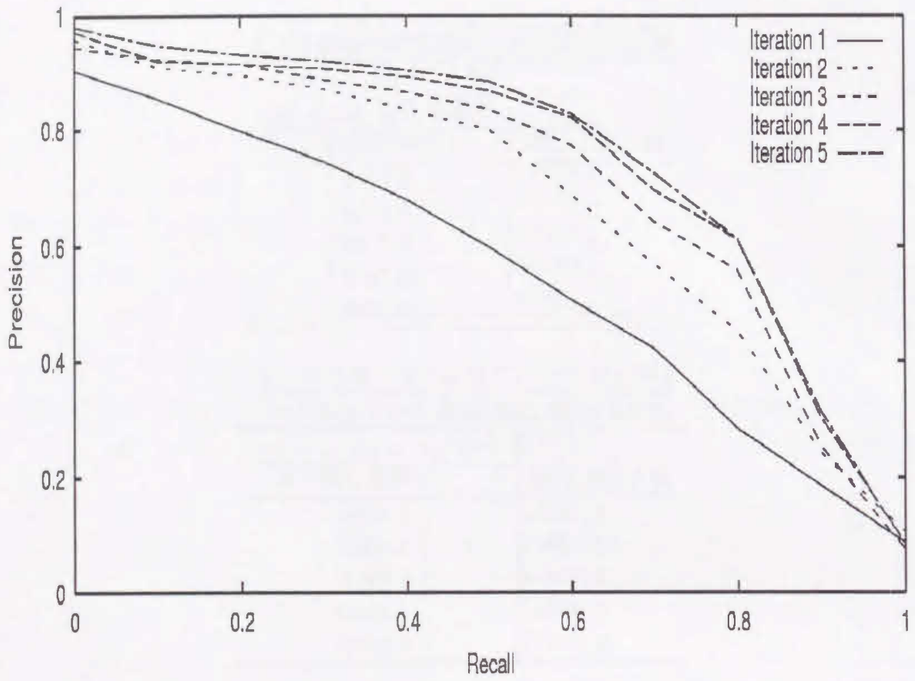

図 5.1 表 5.7における再現率-適合率曲線

ラスタリングして得られた概念ベクトルが，より検索質問が必要としている概念ベクトルに更新されていると考えられる．

より細かく各フィードバック手法を比較すると，関連のある文書を概念ベクトルに最も有効に更新したのは手法 4 で，個々の文書ベクトルを概念ベクトルに反映させるのではなく，検索された関連文書に対する概念ベクトルをフィードバックした方がより効果的であった．また，この手法はフィードバックの効果が早く，3 回程度の学習で平均適合率が最も大きくなっている．このことは，1 回のフィードバックの影響が大きく，さらに，概念ベクトルが検索質問に含まれる概念的な内容に大きく近づいているためであると考えることができる．

しかし，関連のある文書を手法 4 で更新し，さらに，上位に検索されたが検索質問と関連のない文書を概念ベクトルにフィードバックした場合，表 5.4から表 5.6に示したように，何もフィードバックしなかった場合とほとんど同じ結果となった．関連のない文書集合には，それぞれ内容的に関連性のない文書が存在する可能性があるため，手法 1 や手法 2 のようにひとつの文書を概念ベクトルに反映させた．しかし，ひとつの文書を概念ベクトルに反映させた場合，ひとつの文書ベクトルがスパースなベクトルであるためか，フィードバックの効果が少ない結果となった．

これらの手法に対する有効性を比較するために，最も一般的なフィードバック手法である手法 5 を用いて実験した結果，表 5.6に示すように適合率は約 0.96 となった．この手法と比較した場合，我々の提案した手法は良い検索結果を得ることができなかった．しかし，本手法と同様にシステムに対してフィードバックを加える文献 [41] で提案された手法と比較すると，同じ条件の下において 0.7019 を上回る検索結果を得ることができた．システムに対してフィードバックを行う場合，関連のない文書が前回同様に上位に検索されないように，どのようにシステムのパラメータを更新するかが問題となる．本実験では，関連のない文書を概念ベクトルにフィードバックしたときの影響が少なかったため，これをより有効にフィードバックする手法を考慮する必要であると考えられる．これに実現することで，手法 5 のような検索質問拡張の効果に匹敵する検索性能が得られるのではないかと予想される．

また，手法 6に示すように，手法 5 で検索質問に直接フィードバックを行った後，コンセプト・プロジェクションにより次元圧縮をした結果，表 5.7や 5.8に示すように，検索性能は概念ベクトルを更新する手法と比較して，あまり良い結果とはならなかった．このことは，検索質問が更新されたとしても，概念ベクトルはそのまま同じであるために，次元圧縮を行っても，ひとつの特徴軸に対してあまり大きな変化が認められなかった．すなわち，フィードバックによりある単語が拡張されたとしても，次元圧縮時においてその単語の影響が少なくなってしまった結果であると考えられる．

5.5　結言

本稿では，次元圧縮手法であるコンセプト・プロジェクションに必要な概念ベクトルのパラメータをフィードバックさせる手法を提案し，検索精度の改善を試みた．その結果，最も一般的なフィードバック手法を用いて実験し，この手法と比較した結果，我々の提案した手法はそれより良い検索結果を得ることができなかった．しかし，検索質問拡張のような検索時のみの短期的な効果ではなく，システムに対してフィードバックを加え，システムの検索性能を長期的に高める手法としては，本手法の有効性を示すことができた．

今後の課題としては，本実験で用いたテスト・コレクションである MEDLINE には収録されているデータの少なく，さらに医学関連という特定の分野のデータであるため専門用語などの単語が強く影響を与える可能性が高いことがが問題となる．このため，日本語情報検索の評価用テスト・コレクションである BMIR-J2[20] などのような大規模で，様々な分野の内容を持ったテスト・コレクションを用いた検索実験，および評価をすることが必要であると考えている．また，検索質問と関連のある，および関連のないそれぞれの文書集合に対して，これまでに示したフィードバック手法を改良し，より概念ベクトルの性能，および検索性能の改善を行いたいと考えている．

第 6 章　結論

以上，本論文ではベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究として，ベクトル空間モデルを中心に，現在までに行われてきた情報検索手法の研究ついて議論を行った．特に，第 2 章では，単語の意味や共起関係などの情報を用いて検索を行う手法や，ベクトル空間の構造を利用してベクトルの次元を圧縮する手法として有効な，LSI (Latent Semantic Indexing) を紹介した．

第 3 章では，IREX ワークショップにおける IR の本試験の結果，および，参加したすべての情報検索システムについてのアンケートを基に，平均適合率，再現率・適合率曲線を直線回帰させた傾きと切片が，情報検索システムに用いられた手法とどのような相関関係をもっているのかを調査し，それぞれの手法がシステムの性能に与える影響の大きさを示した．また，NARRATIVE タグの使用有無により short と long に分け，平均適合率との相関関係，また，再現率・適合率曲線を直線回帰させた傾きと切片にそれぞれどのような相関関係があるかを調査し，システムの性能に与える影響の大きさを示した．その結果，名詞や名詞以外の品詞単語を用いる以上にフレーズが性能向上に関係あり，複数の単語を組み合わせることで意味が限定され，精度に良い影響を与えることを確認することができた．また，再現率・適合率曲線の切片と傾きにトレードオフの関係が多くの手法に見られ，検索システムに用いる手法の選択の難しさが現れる結果となった．

第 4 章では，LSI の問題点を解決するために，ランダム・プロジェクションを用いた情報検索モデルを構築し，情報検索における次元圧縮手法として，ランダム・プロジェクションの有効性を確認した．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルに，文書の内容を表す概念ベクトル [10] の利用し，これまで単語などが要素であったベクトルを文書の内容を要素とする低次元のベクトルに変換をするコンセプト・プロジェクションを提案した．

さらに，第 5 章では，提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法を提案した．本手法により，更新された概念ベクトルから検索要求や検索対象となる文書ベクトルの次元圧縮が行われるため，フィードバック学習の影響が検索要求だけでなく検索対象にも反映させることができた．

本研究テーマに関する今後の課題は，より効率のよいフィードバック方法の提案が挙げ

られる．最も一般的なフィードバックモデルと比較しても，コンセプト・プロジェクションを用いたフィードバックの効果が少なかった．そこで，検索結果における適合していない文書集合を再度検索しないようにより効率よくフィードバックを行う必要がある．また，概念ベクトルの直交性と独立性の差が検索精度にどれほど影響があるのかを研究する予定である．

謝辞

本研究の全課程を通じ，直接懇切，丁寧な御指導，御鞭撻を賜わった徳島大学工学部知能情報工学科基礎情報工学講座 北 研二教授に心より感謝を申し上げます．

本研究を行うにあたり，絶えず熱心に御指導，御教授を賜わった徳島大学工学部知能情報工学科知能工学講座 青江 順一教授，矢野 米雄教授に心から感謝を申し上げます．

本研究を行う機会を与えてくださり，有用なデータを提供して下さった IREX ワークショップ実行委員の方々，及び，IREX-IR の本試験に参加した方々と判定者の方々に心から感謝を申し上げます．

また，これまでの研究活動において，数多くの助言やサポートをして下さった徳島大学総合科学部国際経済社会システムコース 眞弓 浩三教授，徳島大学工学部知能情報工学科基礎情報工学講座 獅々堀 正幹講師，徳島大学工学部知能情報工学科基礎情報工学講座 石田 富士雄技官，兵庫大学経済情報学部 田中 康仁教授，ならびに，吹谷 和雄社長をはじめとする意味解析技術応用研究所株式会社の皆様に心から感謝を申し上げます．

修了，卒業をしても，絶えず熱心にご助言を頂いた，株式会社ワイ・ディ・シー芦辺 和彦氏をはじめとする北研究室の諸先輩方，NTT ソフトウェア 小田 裕樹氏，富士通株式会社 山下 高史氏，日立システムアンドサービス 守屋 知佐子氏をはじめとする北研究室の修了生，その他，徳島大学知能情報工学科基礎情報工学講座 A-2 グループの諸氏，ならびに，数多くのアイデアや助言を頂いた徳島大学大学院工学研究科 柘植 覚氏，岡田 真氏に心から感謝を申し上げます．

ここに記して，以上の方々に深く感謝の意を表します．

参 考 文 献

[1] 相澤彰子. 語と文書の共起に基づく特徴度の数量的表現について. 情報処理学会論文誌, Vol. 41(12), pp. 3332–3343, 2000.

[2] 新谷研, 角田達彦, 大石巧, 長尾真. 形態素の共起頻度と出現位置による新聞関連記事の検索手法. 信学技報, 電子情報通信学会, 1995.

[3] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Foundations of Computer Science*, pp. 616–623, 1999.

[4] M. W. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval.* SIAM Book Series: Software, Environments, and Tools, 1999.

[5] M. W. Berry, Z. Drmac, and E.R. Jessup. Matrices, vector spaces, and information retrieval. In *SIAM Review*, Vol. 42(2), pp. 335–362, 1999.

[6] A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. In *Proceedings of the 30th ACM Symposium on the Theory of Computing*, pp. 90–99, 1998.

[7] C. Buckley, A. Singhal, M. Mitra, and G. Salton. New retrieval approaches using smart: TREC 4. In *D. Harman (Ed.), Overiew of The Third Text REtrieval Conference (TREC4) National Institute of Standards and Technology Special Publication*, pp. 25–48, 1996.

[8] E. Chicholm and T. G. Kolda. New term weighting formulas for the vector space method in information retrieval. Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1998.

[9] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, Vol. 41(6), pp. 391–407, 1990.

[10] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. Technical report, IBM Almaden Research Center, 1999.

[11] S. T. Dumais. Improving the retrieval of information from external sources. In *Behavior Research Methods, Instruments, & Computers*, Vol. 23, pp. 229–236, 1991.

[12] S. T. Dumais. Using lsi for information filtering: TREC-3 experiments. In *D. Harman (Ed.), Overiew of The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication*, pp. 219–230, 1995.

[13] C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pp. 163–174, 1995.

[14] U. Feige. Approximating the bandwidth via volume respecting embeddings. In *Proceedings of the 30th ACM Symposium on the Theory of Computing*, 1998.

[15] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Combinatorial Theory*, Vol. B 44, pp. 355–362, 1988.

[16] D. Harman. Ranking algorithms, in information retrieval: Data structures & algorithms. In *W. B. Frakes and R. Baeza-Yates (Eds.), Information Retrieval, Prentice Hall, NJ*, pp. 363–392, 1992.

[17] D. Hull. Using statistical testing in the evaluation of retrieval performance. In *Proceedings of ACM SIGIR Conference*, pp. 329–338, 1993.

[18] D. Hull. Stemming algorithms — a case study for detailed evaluation. *Journal of the American Society for Information Science*, Vol. 47(1), pp. 70–84, 1996.

[19] W. B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into hilbert space. *Contemporary Mathematics*, Vol. 26, pp. 189–206, 1984.

[20] 木谷強ほか. 日本語情報検索システム評価用テストコレクション BMIR–J2. データベースシステム研究会, 情報処理学会, 1998.

[21] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Segmentation problems: A microeconomic view of data mining. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pp. 473–482, 1998.

[22] 国立国語研究所. 新分類語彙表. Technical report, 国立国語研究所, 1996.

[23] T. G. Kolda and D. P. O'Leary. A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. In *Proceedings of ACM Transaction on Information Systems*, Vol. 16, pp. 322–346, 1998.

[24] E. Lagergren and P. Over. Comparing interactive information retrieval systems across

sites: the TREC-6 interactive track matrix experiment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 164–172, 1998.

[25] D. D. Lewis. Representation quality in text classification: An introduction and experiment. In *Defense Advanced Research Projects Agency*, pp. 288–295, 1990.

[26] D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pp. 312–318, 1991.

[27] 松本裕治, 今一修, 山下達雄, 北内啓, 今村友明. 日本語形態素解析システム「茶筅」使用説明書. Naist technical report, 奈良先端科学技術大学院大学, 1997.

[28] 松尾衛, 宮本昌幸, 森辰則. 情報検索タスクにおける人間による正解判定についての分析. In *Proceedings of the IREX workshop*, pp. 15–22, 1999.

[29] 西野文人. 日本語テキスト分類における特徴素抽出. 自然言語処理研究会, 情報処理学会, 1996.

[30] C. D. Paice. Another stemmer. *SIGIR Forum*, Vol. 24, pp. 56–61, 1990.

[31] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the 17th ACM Symposium on the Principles of Database Systems*, pp. 159–168, 1998.

[32] M. F. Porter. An algorithm for suffix stripping. *Program*, Vol. 14(3), pp. 130–137, 1980.

[33] Edie Rasmussen. *Clustering Algorithms.* (W.B. Frakes and R. Baeza-Yates, eds.). Englewood Cliffs, NJ: Prentice Hall, 1991.

[34] J. J. Rocchio. Relevance feedback in information retrieval. In *Salton G. (Ed.), The SMART Retrieval System. Englewood Cliffs, N.J.: Prentice Hall*, pp. 313–323, 1971.

[35] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Journal of Information Processing and Management*, Vol. 24(5), pp. 513–523, 1988.

[36] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, Vol. 41, pp. 288–297, 1990.

[37] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[38] 関根聡, 井佐原均. IREXプロジェクト概要. In *Proceedings of the IREX workshop*, pp. 1–5, 1999.

[39] A. Singhal, C. Buckley, M. Mitra, and G. Salton. Pivoted document length normalization. Technical report, Department of Computer Science, Cornell University, Ithaca, New York, 1995.

[40] X. Tai and K. Kita. 教師あり学習によるベクトル空間モデルの精度改善. 自然言語処理研究会, 情報処理学会, 2000.

[41] Xiao-Ying Tai, Minoru Sasaki, Kenji Kita, and Yasuhito Tanaka. Improvement of vector space information retrieval model based on supervised learning. In *Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages(IRAL2000)*, pp. 69–74, 2000.

[42] T.Kohonen 著, 徳高平蔵, 岸田悟, 藤村喜久郎訳. 自己組織化マップ. シュプリンガー・フェアラーク東京, 1996.

[43] 徳永健伸. 情報検索と言語処理. 東京大学出版会, 1999.

[44] Iwayama Makoto Tokunaga Takenobu. Text categorization based on weighted inverse document frequency. Technical report, Dept. of Computer Science Tokyo Institute of Technology, 1994.

[45] S. Vempala. Random projection: A new approach to vlsi layout. In *Proceedings of the 39th Foundations of Computer Science, Palo Alto*, pp. 389–395, 1998.

[46] C. C. Vogt, G. W. Cottrell, R. K. BeLew, and B. T. Bartell. User lenses - achieving 100% precision on frequently asked questions. In *Proceedings of User Modeling '99, Banff*, pp. 87–96, 1999.

[47] E. M. Voorhees. Variations in relevance judgements in the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 315–323, 1998.

[48] E. M. Voorhees and D. Harman. Overview of the sixth text retrieval conference (TREC-6). In *Voorhees and Harman (Eds.), Proceedings of The Sixth Text REtrieval Conference (TREC-6) National Institute of Standards and Technology Special Publication*, pp. 1–24, 1997.

[49] E. M. Voorhees and D. Harman. Overview of the seventh text retrieval conference (TREC-7). In *Voorhees and Harman (Eds.), Proceedings of The Seventh Text RE-*

trieval Conference (TREC-7) National Institute of Standards and Technology Special Publication, pp. 1–24, 1998.

[50] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Van Nostrand Reinhold, New York, 1994.

付録 A　IR システムアンケート

0. システム ID

0.1. IREX 固有の情報

0.1.1. 使用した検索課題中の情報（DESCRIPTION のみ (D)、NARRATIVE のみ (N)、DESCRIPTION と NARRATIVE 両方 (D& N))

0.1.2. NEG のタグは利用しましたか？

0.1.3. システム開発に関連して過去に９４、９５年毎日新聞を使用したことがありますか？

0.1.4. ９５年８月２４、２５日のデータのバグについて予め対処してありますか？

1. 索引づけ

1.1. 索引づけに用いた方法

1.1.1 日本語の索引単位は何か？(uni-gram,bi-gram, その他の n-gram, 単語, フレーズ, その他)

1.1.2. どのような方法を用いて索引づけをしたか？(辞書、形態素解析、その他)

1.1.3. 索引語の選択方法は何か？(例：ストップワード、字種、品詞など)

1.1.4. 語彙（文字）の正規化を行なったか？

1.1.5. ステミングアルゴリズムを用いたか？

1.1.6. 語の重みづけを用いたか？

1.1.7. フレーズ単位で索引づけをしたか？

1.1.8. フレーズの種類は？

1.1.9. フレーズの見つけ方は？(統計的、構文的、その他)

1.1.10. 構文解析は行なったか？

1.1.11. シソーラスや用語集などを用いたか？

1.1.12. 語義の曖昧性解消は行なったか？

1.1.13. 誤字脱字やスペルのチェックは行なったか？

1.1.14. 誤字脱字やスペルの修正は行なったか？

1.1.15. 固有名詞を識別したか？

1.1.16. どのような方法で索引単位に分割したか？

1.1.17. 日本語のヨミを用いたか？

1.1.18. ヨミを用いた場合、ヨミはどのように生成したか？

1.1.19. 索引づけに用いたその他の方法（具体的に）

1.2. 毎日新聞から構築された索引データの構造

1.2.1. 索引の構造の種類

1.2.1.1. クラスタ

1.2.1.2. シグネチャファイル

1.2.1.3. Pat 木

1.2.1.4. 知識ベース

1.2.1.5. その他（具体的に）

1.2.2. 索引の概要

1.2.2.1. 索引の規模 [MB]

1.2.2.2. 構築に要した時間［時間]

1.2.2.3. 実行過程は自動化されているか？

1.2.2.4. 語の出現位置（オフセット）は使用したか？

1.3. 毎日新聞以外の情報源から索引作成のために利用したデータ

1.3.1. 独自に構築したデータ（種類＝シソーラス、知識ベース、辞書など）

1.3.2. 外部で構築されたデータ（種類とデータ名）

2. 検索式の作成

2.1. 検索式を作成するのに要した時間（１課題当たりの平均 CPU 時間［秒]）

2.2. 検索式作成に使用した方法

2.2.1. 索引単位への分割（uni-gram,bi-gram, その他の n-gram, 単語, フレーズ, その他）

2.2.2. フレーズの抽出

2.2.3. 構文解析

2.2.4. 語義の曖昧性解消

2.2.5. 固有名詞の識別

2.2.5. 検索式の自動拡張

2.2.5.1. シソーラスなど既存のツール

2.2.5.2. 自動レレバンスフィードバック

2.2.5.3. ローカルコンテクストアナリシス

2.2.5.4. その他（具体的に）

2.2.6. ブール演算子や近接演算子などの自動的付与

2.2.7. その他（具体的に）

3. 検索実行

3.1. 検索時間（１検索式に対する平均 CPU 時間［秒]）

3.2. プロセスサイズ [MB]

3.3. 計算機についての情報

3.3.1. 実験に使用した計算機

3.3.2. その計算機は専用か共用か

3.3.3. ハードディスクの総容量 [GB]

3.3.4. RAM の総容量 [MB]

3.3.5. CPU のクロック数 [MHz]

4. 検索モデル

4.1. ベクトル空間型を用いたか？

4.2. 確率型を用いたか？

4.3. その他（具体的に）

4.4. ランクづけの要素

4.4.1. TF（語の出現頻度）を使用したか？

4.4.2. IDF を使用したか？

4.4.3. その他の重みづけ（具体的に）

4.4.4. 意味の近さを使用したか？

4.4.5. 文書中の位置を使用したか？

4.4.6. 構文的な手がかりを使用したか？

4.4.7. 語の近接（距離）を使用したか？

4.4.8. 文書の長さを使用したか？

4.4.9. その他（具体的に）

5. その他

5.1. 上の質問で回答していないシステムの特色

5.2. チームの構成員

5.2.1. 日本語を母国語とする人がいますか？

5.2.2. 日本語のわかる人がいますか？

5.3. 関連データの利用

5.3.1. BMIR-1 を利用しましたか？

5.3.2. BMIR-2 を利用しましたか？

5.3.3. NACSIS-collection を利用しましたか？

5.3.4. IREX-IR 予備試験データを利用しましたか？

5.3.5. TREC データを利用しましたか？

付録 B アンケート回答

system ID	1103a	1103b	1106	1110
0.1.1.	D	D	D& N	D
0.1.2.	×	×	○	×
0.1.3.	BMIR-J2 の経験あり	BMIR-J2 の経験あり	×	×
0.1.4.	×	×	×	×
1.1.1.	単語および n-gram	単語および n-gram	単語	単語
1.1.2.	形態素解析など	形態素解析など	形態素解析	辞書
1.1.3.	品詞など	品詞など	品詞	品詞
1.1.4.	×	×	×	○
1.1.5.	×	×	×	○○
1.1.6.	○	○	○	○
1.1.7.	○	○	×	×
1.1.8.	?	?	-	-
1.1.9.	形態素解析	形態素解析	-	-
1.1.10.	×	×	×	×
1.1.11.	×	×	×	×
1.1.12.	×	×	×	×
1.1.13.	×	×	×	×
1.1.14.	×	×	×	×
1.1.15.	○	○	×	×
1.1.16.	形態素解析など	形態素解析など	形態素解析	最長一致
1.1.17.	×	×	×	×
1.1.18.	-	-	-	-
1.1.19.	-	-	-	-
1.2.1.1.	×	×	-	-
1.2.1.2.	×	×	-	-
1.2.1.3.	×	×	-	-
1.2.1.4.	×	×	-	-
1.2.1.5.	A1	A2	転置ファイル	A3
1.2.2.1.	?	?	38MB	?
1.2.2.2.	?	?	?	?
1.2.2.3.	○	○	○	○
1.2.2.4.	○	○	×	×
1.3.1.	既存の形態素解析辞書	既存の形態素解析辞書	-	単語のベクトル辞書
1.3.2.	-	-	形態素解析システム、茶筅	-
2.1.	1 秒以内	1 秒以内	0.002 秒	?
2.2.1.	単語	単語	単語	単語
2.2.2.	×	×	-	×
2.2.3.	×	×	-	×
2.2.4.	×	×	-	×
2.2.5.	×	×	-	×
2.2.5.	×	×	-	×
2.2.5.1.	×	×	-	×
2.2.5.2.	×	○ A4	-	-
2.2.5.3.	×	×	-	-
2.2.5.4.	-	-	-	-
2.2.6.	×	×	-	×
2.2.7.	-	-	A5	-
3.1.	?	?	8 秒	7.27 秒
3.2.	?	?	?	?
3.3.1.	SUN Ultra 2	SUN Ultra 2	Pro2333DL	Pentium 2
3.3.2.	専用	専用	共用	専用
3.3.3.	23GB	23GB	6.4GB	8GB
3.3.4.	768MB	768MB	192MB	128MB
3.3.5.	296MHz	296MHz	333MHz	300MHz
4.1.	○	○	×	×
4.2.	○	○	-	×
4.3.	-	-	-	-
4.4.1.	○	○	○	○
4.4.2.	○	○	○	○
4.4.3.	BM25	BM25	-	-
4.4.4.	×	×	×	○
4.4.5.	×	×	×	×
4.4.6.	×	×	×	×
4.4.7.	×	×	×	×
4.4.8.	○	○	○	×
4.4.9.	-	-	-	-
5.1.	-	-	-	-
5.2.1.	○	○	○	○
5.2.2.	○	○	○	○
5.3.1.	×	×	×	○
5.3.2.	×	×	×	○
5.3.3.	×	×	○	×
5.3.4.	○	○	○	○
5.3.5.	×	×	×	×

system ID	1112	1120	1122a	1122b
0.1.1.	D	D& N	D& N	D& N
0.1.2.	×	○	○	○
0.1.3.	×	×	×	×
0.1.4.	○	×	○	○
1.1.1.	単語	単語	単語	単語
1.1.2.	形態素解析	形態素解析	辞書を用いた極大単語切り出し	辞書を用いた極大単語切り出し
1.1.3.	品詞	名詞	ストップワード以外を選択	ストップワード以外を選択
1.1.4.	?	×	×	×
1.1.5.	○	×	×	×
1.1.6.	○	○	○	○
1.1.7.	○	×	×	×
1.1.8.	単語	-	×	×
1.1.9.	構文的	-	-	-
1.1.10.	○	×	×	×
1.1.11.	×	×	×	×
1.1.12.	×	×	×	×
1.1.13.	×	×	×	×
1.1.14.	×	×	×	×
1.1.15.	×	×	×	×
1.1.16.	形態素解析	形態素解析	極大単語索引方式	極大単語索引方式
1.1.17.	×	×	×	×
1.1.18.	×	-	-	-
1.1.19.	×	-	-	-
1.2.1.1.	?	-	-	-
1.2.1.2.	×	-	-	-
1.2.1.3.	A6	-	-	-
1.2.1.4.	×	-	-	-
1.2.1.5.	A7	-	極大単語索引	極大単語索引
1.2.2.1.	150MB	225MB	300MB	300MB
1.2.2.2.	?	○	○	○
1.2.2.3.	○	○	○	○
1.2.2.4.	○	×	○	○
1.3.1.	辞書	-	辞書	辞書
1.3.2.	EDR 日本語単語辞書	-	EDR 辞書	EDR 辞書
2.1.	?	10 秒	4.3 秒	0.3 秒
2.2.1.	単語	単語	単語	単語
2.2.2.	構文解析	-	×	×
2.2.3.	A8	-	×	×
2.2.4.	×	-	×	×
2.2.5.	×	-	×	×
2.2.5.	×	-	×	×
2.2.5.1.	×	-	○	○
2.2.5.2.	×	-	×	×
2.2.5.3.	×	-	○	○
2.2.5.4.	×	-	×	-
2.2.6.	○	-	A9	○
2.2.7.	検索要求の係り受け関係を使用	-	-	-
3.1.	検索式作成を含め 4.23 秒	2500 秒	0.6 秒	0.6 秒
3.2.	0.2MB	0.6MB	0.5MB	0.5MB
3.3.1.	DELL Dimension R350	AS7000 U1/140	SUN SS-UA2	SUN SS-UA2
3.3.2.	専用	共用	共用	共用
3.3.3.	12GB	6GB	64GB	64GB
3.3.4.	192MB	64MB	64MB	64MB
3.3.5.	350MHz	140MHz	296MHz	296MHz
4.1.	×	○	○	○
4.2.	×	×	×	×
4.3.	文書の係り受け関係を使用	-	-	-
4.4.1.	○	-	○	○
4.4.2.	○	-	○	○
4.4.3.	A10	WIDF	文書内共起情報を利用	文書内共起情報を利用
4.4.4.	×	×	×	×
4.4.5.	×	×	×	×
4.4.6.	○	×	×	×
4.4.7.	○	×	×	×
4.4.8.	×	×	○	○
4.4.9.	語の係り受け	タイトル中の単語共起	-	-
5.1.	-	-	-	-
5.2.1.	○	○	○	○
5.2.2.	○	○	○	○
5.3.1.	×	×	×	×
5.3.2.	○	×	評価データとして利用した	評価データとして利用した
5.3.3.	×	×	間接的に利用した	間接的に利用した
5.3.4.	○	○	評価データとして利用した	評価データとして利用した
5.3.5.	×	×	×	×

system ID	1126	1128a	1128b
0.1.1.	D	D& N	D& N
0.1.2.	×	×	×
0.1.3.	×	×	○
0.1.4.	○	×	×
1.1.1.	単語と修飾情報	その他の n-gram	全部分文字列
1.1.2.	A11	その他	Suffix Array
1.1.3.	品詞	A11	全部分文字列をすべて利用
1.1.4.	×	×	×
1.1.5.	×	×	×
1.1.6.	○	○	×
1.1.7.	×	○	×
1.1.8.	×	情報検査の情報料できまるもの	-
1.1.9.	×	統計的	-
1.1.10.	○	×	×
1.1.11.	×	×	×
1.1.12.	×	×	×
1.1.13.	×	×	×
1.1.14.	×	×	×
1.1.15.	○	×	×
1.1.16.	形態素解析	最大スコアパスを与える n-gram	全部分文字列(Suffix Array)
1.1.17.	×	×	×
1.1.18.	×	-	-
1.1.19.	×	-	-
1.2.1.1.	?	-	-
1.2.1.2.	?	-	-
1.2.1.3.	?	-	-
1.2.1.4.	?	-	-
1.2.1.5.	?	suffix array は使用した	Suffix Array
1.2.2.1.	1.3GB	600MB	412MB
1.2.2.2.	87 時間	2 時間	1 時間
1.2.2.3.	○	○	○
1.2.2.4.	○	×	○
1.3.1.	A3	×	×
1.3.2.	×	×	×
2.1.	?	0 秒	?
2.2.1.	単語	その他の n-gram	全部分文字列
2.2.2.	×	×	-
2.2.3.	A4	×	-
2.2.4.	×	×	-
2.2.5.	固有名詞辞書	×	-
2.2.5.	×	×	-
2.2.5.1.	×	×	-
2.2.5.2.	×	×	-
2.2.5.3.	×	×	-
2.2.5.4.	×	×	-
2.2.6.	×	×	-
2.2.7.	×	文字列のまま使用	A15
3.1.	294 秒	1800 秒	600 秒
3.2.	?	IGB	1000MB
3.3.1.	Sun Ultra1	Pentium 2	SUN Enterprise 3000
3.3.2.	共用	専用	専用
3.3.3.	4GB	24GB	50GB
3.3.4.	256MB	512MB	1500MB
3.3.5.	UltraSPARC 2 300MHz x 2 units	400MHz	200MHz
4.1.	○	×	○
4.2.	×	×	×
4.3.	×	A16	-
4.4.1.	○	×	○
4.4.2.	○	○	○
4.4.3.	×	IDF のみ	term のみ
4.4.4.	○	×	×
4.4.5.	○	×	×
4.4.6.	○	×	×
4.4.7.	×	×	×
4.4.8.	○	×	○
4.4.9.	×	-	-
5.1.	×	A17	A18
5.2.1.	○	○	○
5.2.2.	○	○	○
5.3.1.	×	×	×
5.3.2.	×	×	×
5.3.3.	○	○	○
5.3.4.	間接的に使用した	×	○
5.3.5.	×	×	×

system ID	1132	1133a	1133b	1135a
0.1.1.	D	D& N	D& N	D& N
0.1.2.	×	×	×	○
0.1.3.	×	○	○	×
0.1.4.	×	×	×	×
1.1.1.	単語	その他	その他	×
1.1.2.	形態素解析	その他	その他	×
1.1.3.	品詞	-	-	×
1.1.4.	×	×	×	×
1.1.5.	×	×	×	×
1.1.6.	○	○	○	×
1.1.7.	×	×	×	×
1.1.8.	-	×	×	×
1.1.9.	-	その他	その他	×
1.1.10.	×	×	×	×
1.1.11.	○	×	×	×
1.1.12.	×	×	×	×
1.1.13.	×	×	×	×
1.1.14.	×	×	×	×
1.1.15.	×	×	×	×
1.1.16.	形態素解析	-	-	×
1.1.17.	×	×	×	×
1.1.18.	-	-	-	×
1.1.19.	Latent Semantic Indexing	-	-	×
1.2.1.1.	-	×	×	×
1.2.1.2.	-	×	×	×
1.2.1.3.	-	×	×	×
1.2.1.4.	-	×	×	×
1.2.1.5.	-	-	-	×
1.2.2.1.	418MB	-	-	×
1.2.2.2.	48 時間	-	-	×
1.2.2.3.	×	×	×	×
1.2.2.4.	×	×	×	×
1.3.1.	形態素解析用専用辞書	-	-	×
1.3.2.	-	-	-	×
2.1.	2 秒	1 秒	1 秒	60 秒
2.2.1.	単語	-	-	単語
2.2.2.	-	-	-	×
2.2.3.	-	-	-	×
2.2.4.	-	-	-	×
2.2.5.	-	-	-	○
2.2.5.	A19	-	-	×
2.2.5.1.	-	-	-	×
2.2.5.2.	-	-	-	×
2.2.5.3.	-	-	-	×
2.2.5.4.	-	-	-	×
2.2.6.	-	-	-	×
2.2.7.	-	-	-	×
3.1.	74 秒	1200 秒	1200 秒	1200 秒
3.2.	453MB	5MB	5MB	500MB
3.3.1.	アルファーサーバー 4000	Gateway 2000	Gateway 2000	Sun Ultra 10
3.3.2.	専用	専用	専用	専用
3.3.3.	40GB	8GB	8GB	1000GB
3.3.4.	1000MB	?	?	1000MB
3.3.5.	?	180MHz	180MHz	?
4.1.	○	×	×	×
4.2.	×	○	○	?
4.3.	-	-	-	ロバートソン式の変換
4.4.1.	○	○	○	○
4.4.2.	×	○	○	○
4.4.3.	-	「面」の情報を使った	「面」の情報を使った	-
4.4.4.	×	×	×	×
4.4.5.	×	×	×	○
4.4.6.	×	×	×	○
4.4.7.	×	×	×	×
4.4.8.	×	×	×	○
4.4.9.	-	-	-	×
5.1.	-	-	-	A20
5.2.1.	○	○	○	○
5.2.2.	○	○	○	○
5.3.1.	×	×	×	×
5.3.2.	×	×	×	×
5.3.3.	×	×	×	○
5.3.4.	○	○	○	○
5.3.5.	×	×	×	×

system ID	1135b	1142	1144a	1144b
0.1.1.	D& N	D	D& N	D& N
0.1.2.	○	×	○	○
0.1.3.	×	×	○	○
0.1.4.	×	×	○	○
1.1.1.	×	その他の n-gram	-	-
1.1.2.	×	その他	形態素解析	形態素解析
1.1.3.	×	洗濯せずに全て使用	ストップワード、字種、品詞	ストップワード、字種、品詞
1.1.4.	×	○	○	○
1.1.5.	×	×	×	×
1.1.6.	×	×	○	○
1.1.7.	×	×	○	○
1.1.8.	×	-	-	-
1.1.9.	×	-	-	-
1.1.10.	×	×	×	×
1.1.11.	×	×	○	○
1.1.12.	×	×	×	×
1.1.13.	×	×	×	×
1.1.14.	×	×	○	○
1.1.15.	×	×	×	×
1.1.16.	×	字種ごとに n-gram に分割	形態素解析	形態素解析
1.1.17.	×	×	×	×
1.1.18.	×	×	-	-
1.1.19.	×	-	-	-
1.2.1.1.	×	-	-	-
1.2.1.2.	×	-	-	-
1.2.1.3.	×	-	-	-
1.2.1.4.	×	-	-	-
1.2.1.5.	×	B-tree	転置ファイル	転置ファイル
1.2.2.1.	×	?	164MB	164MB
1.2.2.2.	×	?	1.4 時間	1.4 時間
1.2.2.3.	×	○	○	○
1.2.2.4.	×	○	○	○
1.3.1.	×	×	シソーラス	シソーラス
1.3.2.	×	×	-	-
2.1.	60 秒	?	-	-
2.2.1.	単語	単語	-	-
2.2.2.	×	×	○	○
2.2.3.	×	×	×	×
2.2.4.	×	×	×	×
2.2.5.	○	×	○	○
2.2.5.	×	×	○	○
2.2.5.1.	×	×	○	○
2.2.5.2.	×	×	○	○
2.2.5.3.	×	×	×	×
2.2.5.4.	×	-	-	-
2.2.6.	×	検索語を OR と結合	×	×
2.2.7.	×	-	-	-
3.1.	900 秒	?	7.8 秒 A23	3.4 秒 A24
3.2.	500MB	?	-	-
3.3.1.	Sun Ultra 10	?	DellOptiPlex GX1	DellOptiPlex GX1
3.3.2.	専用	?	専用	専用
3.3.3.	100GB	?	9GB	9GB
3.3.4.	1000MB	?	384MB	384MB
3.3.5.	?	?	450MHz	450MHz
4.1.	×	○	○	○
4.2.	?	×	○	×
4.3.	ロバートソン式の変換	-	-	-
4.4.1.	○	○	○	○
4.4.2.	○	○	○	○
4.4.3.	-	-	-	-
4.4.4.	×	×	×	×
4.4.5.	○	×	×	×
4.4.6.	○	×	×	×
4.4.7.	×	×	×	×
4.4.8.	○	○	×	×
4.4.9.	×	-	-	-
5.1.	誌面情報の利用	-	-	-
5.2.1.	○	○	○	○
5.2.2.	○	○	○	○
5.3.1.	×	○	×	×
5.3.2.	×	○	○	○
5.3.3.	○	×	○	○
5.3.4.	○	×	○	○
5.3.5.	×	×	×	×

system ID	1126	1128a	1128b
0.1.1.	D& N	D& N	D
0.1.2.	×	×	×
0.1.3.	○	○	×
0.1.4.	×	×	×
1.1.1.	単語	単語	uni-gram
1.1.2.	形態素解析	形態素解析	字句間距離
1.1.3.	品詞、ストップワード	品詞、ストップワード	字種
1.1.4.	○	○	○
1.1.5.	×	×	×
1.1.6.	○	○	○
1.1.7.	×	×	×
1.1.8.	-	-	?
1.1.9.	-	-	?
1.1.10.	×	×	×
1.1.11.	×	×	×
1.1.12.	×	×	×
1.1.13.	×	×	×
1.1.14.	×	×	×
1.1.15.	×	×	×
1.1.16.	-	-	文字分解
1.1.17.	×	×	×
1.1.18.	-	-	?
1.1.19.	-	-	字区間距離マップ作成
1.2.1.1.	-	-	×
1.2.1.2.	-	-	×
1.2.1.3.	-	-	×
1.2.1.4.	-	-	×
1.2.1.5.	-	-	字区間距離マップ
1.2.2.1.	390MB	390MB	1043MB
1.2.2.2.	2時間	2時間	3.83時間
1.2.2.3.	○	○	○
1.2.2.4.	×	×	○
1.3.1.	-	-	×
1.3.2.	-	-	×
2.1.	1秒	1秒	?
2.2.1.	単語	単語	uni-gram
2.2.2.	-	-	×
2.2.3.	-	-	×
2.2.4.	-	-	×
2.2.5.	-	-	×
2.2.5.	A25	A26	×
2.2.5.1.	-	-	×
2.2.5.2.	-	-	×
2.2.5.3.	-	-	×
2.2.5.4.	-	-	文字の正規化
2.2.6.	-	-	×
2.2.7.	-	-	×
3.1.	3600秒	3600秒	470秒
3.2.	1000MB	1000MB	0.22MB
3.3.1.	Sun Ultra1 Enterprise 450	Sun Ultra1 Enterprise 450	SUN Ultra10
3.3.2.	共用	共用	共用
3.3.3.	9GB	9GB	12GB
3.3.4.	2560MB	2560MB	128MB
3.3.5.	400MHz	400MHz	300MHz
4.1.	○	○	×
4.2.	×	×	×
4.3.	-	-	照合位置の連続性評価
4.4.1.	○	○	×
4.4.2.	○	○	×
4.4.3.	-	A27	検索文字列との照合文字列長
4.4.4.	×	×	×
4.4.5.	○	○	○
4.4.6.	×	×	×
4.4.7.	×	×	×
4.4.8.	×	×	×
4.4.9.	-	-	検索文字列と照合文字列長
5.1.	-	-	字面の類似性による検索
5.2.1.	○	○	○
5.2.2.	○	○	○
5.3.1.	評価データとして利用した	評価データとして利用した	×
5.3.2.	評価データとして利用した	評価データとして利用した	×
5.3.3.	×	×	×
5.3.4.	評価データとして利用した	評価データとして利用した	○
5.3.5.	×	×	×

様式9

論文審査の結果の要旨

報告番号	甲　工 乙　工　第　193　号 工　修	氏名	佐々木稔
審査委員	主　査　北　研二 副　査　青江順一 副　査　矢野米雄		
学位論文題目	ベクトル空間モデルを用いた情報検索手法の検索精度向上に関する研究		

審査結果の要旨

本論文は，情報検索手法の検索精度向上に関する研究として，情報検索システムに用いられた手法と検索精度に存在する関係の調査と概念ベクトルを用いることにより効率的に次元圧縮を可能とする，情報検索における新しい次元圧縮手法に関する研究の成果をまとめたものである．

第 1 章では，緒論として，情報検索の歴史的背景を述べると共に，本研究の目的ならびにその工学上の意義を述べることで，本研究の意義及び位置付けを明確にしている．第 2 章では，単語の意味や共起関係などの情報を用いて検索を行う手法や，ベクトル空間の構造を利用してベクトルの次元を圧縮する手法として有効な，LSI (Latent Semantic Indexing) について説明している．第 3 章では，IREX ワークショップにおける IR の本試験の結果，および，参加したすべての情報検索システムについてのアンケートを基に，平均適合率，再現率・適合率曲線を直線回帰させた傾きと切片が，情報検索システムに用いられた手法とどのような相関関係をもっているのかを調査し，それぞれの手法がシステムの性能に与える影響の大きさを示した．第 4 章では，LSI の問題点を解決するために，ランダム・プロジェクションを用いた情報検索モデルを構築し，情報検索における次元圧縮手法として，ランダム・プロジェクションの有効性を確認した．また，ランダム・プロジェクションを行う際にあらかじめ指定するベクトルに，文書の内容を表す概念ベクトルの利用し，これまで単語などが要素であったベクトルを文書の内容を要素とする低次元のベクトルに変換をするコンセプト・プロジェクションを提案した．第 5 章では，提案したコンセプト・プロジェクションの応用として，関連性フィードバックによる検索モデルの更新手法を提案した．最後に，第 6 章で本研究で得られた諸成果の総括を行い，今後の研究課題について述べている．

以上本研究は，情報検索システムに用いられた手法の新しい評価方法により，検索システムの設計にに有効な知見を与え，概念ベクトルを用いた効率的な次元圧縮手法であるコンセプト・プロジェクションを提案し，その有効性を考察したものであり，本論文は博士（工学）の学位授与に値するものと判定する．