シソーラスを用いた語の共起関係推定による Rocchio フィードバックの精度向上

中島浩之†

Rocchio フィードバックの検索精度を向上する手段として，決定木学習アルゴリズム ID3 を用いて文書から検索語間の重要な共起関係を抽出し，検索結果に反映させる手法が報告されている．しかし従来手法では質問文中の検索語間の共起のみを扱っていたため，文書中にのみ登場する語を含む共起関係は抽出されなかった．そのため質問文に登場しなければ，検索語の類義語や質問に関連する語であっても，これらを含む共起関係が抽出されることはなかった．本稿ではシソーラスを用いた概念学習により，概念を含む共起関係を抽出することで，類義語を含む共起関係を抽出する．さらに概念分類が粗い場合に重要な共起関係が抽出できない可能性があるため，シソーラスの分類より細かい分類を推定し，共起関係推定の対象とする．提案手法を OHSUMED テストコレクションを用いて評価したところ，検索語の組合せだけではサンプル中の必要文書と不要文書を判別できない場合において，Rocchio フィードバックと比較して上位 10 件での適合率を 10%程度向上させることができた．

Detecting Co-occurence of Concepts with a Thesaurus for Improving Rocchio Feedback

Hiroyuki Nakajima†

It has proposed that using ID3 inductive learning algorithm for capturing co-occurences of query words in order to correct ranked results of the Rocchio feedback. However, the method does not handle synonyms or words relative to queries which appear only in documents, and it could not capture any co-occurences including these words. In this paper, we apply inductive learning algorithm which uses a thesaurus as background knowledge, and find co-occurences including concepts to capture co-occurences including synonyms. Futhermore, to avoid the case that some concepts have too general meanings to distinguish relevant documents from irrevant ones, we propose the method that estimates more specified concepts in finding co-occurences. Experimental results on OHSUMED test collection show that the proposed method improves retrieval accuracy by 10% from Rocchio feedback if any co-occurences of query words are not enough to distinguish relevant documents from irrevant ones in sample documents.

1. はじめに

文書データベースから必要な文書を検索する場合，対象となる文書を正確に表現する検索式を作成する必要がある．しかし正確な検索式を作成するためには，検索対象となる文書の内容について十分な知識が必要であり，必要な文書を検索により入手する前の検索者にとって適切な検索式を作成するのは困難である．レレバンスフィードバックはこの問題を解決する手法であり，システムと検索者が協調して検索式を作成することで，検索者にとって容易かつ高い精度で文書検索を行う手段である．検索者はまず初期の検索式を与え，この検索式により得られた文書から必要文書と不要文書を選択すると，選択された文書からシステムが自動的に検索式を更新し，検索を行う．この選択による検索式の更新がレレバンスフィードバックであり，検索結果について必要文書と不要文書を選択することで，利用者は容易に必要文書を収集することができる．

レレバンスフィードバックを実現する代表的なアルゴリズムである Rocchio フィードバック[1]は，検索要求文および検索対象の文書をベクトルとして表現するベクトル空間法[2]（Vector Space Model）を用いて，文書検索の精度を向上させる手法である．検索者は検索要求文による検索結果の一部について必要か不要かを判断し，システムにフィードバックする．システムはフィードバックされた文書中の単語を用いて再度検索を行う．次に検索要求文を表すベクトルを修正し，

† NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation

検索された各々の文書を表すベクトルとの内積をスコアとして，検索された文書をスコアの高い順に順位付けして呈示する．検索者は高い順位の文書から閲覧するため，システムが必要な文書に高い順位を与えれば，閲覧される文書が必要な文書である可能性が高くなる．つまり検索者が閲覧する文書の集合が高い検索精度を持つことになる．

Rocchio フィードバックは高精度の文書検索を実現する手段として，多くの研究者からその有効性が報告されている[3),4)]が，ベクトルの修正において検索語間の関係は考慮されないため，複数の検索語が 1 つの文書中に現れる（共起する）ことで具体的な内容を指す文書に対しては，適切なスコア付けが行われないことがあった．

筆者はフィードバックされた文書から決定木学習アルゴリズム ID3[5)]を用いて検索語間の重要な共起を推定し，推定した共起を含む文書の順位を上昇させることで，Rocchio フィードバックの検索精度を向上させる手法を提案している[6)]．しかし従来手法では，検索要求文中の検索語間についてのみ共起を推定していたため，以下の語は共起推定の対象とならなかった．

(1) 検索語と類似した意味の語（類義語）
(2) フィードバックされる文書中にのみ登場する，検索要求文に関連する語（関連語）

このため従来手法では重要な共起関係を抽出できない恐れがあり，これらの語を含む共起関係を推定することで，より検索精度を向上できる可能性がある．

シソーラスを参照して検索語と同じ概念に属する語をすべて概念に置換（抽象化）し，概念間での共起関係を抽出することで，上記 (1) にあげた，検索語と同一概念に属する類義語を含む共起関係を抽出することができる．しかしシソーラスの同一概念に属する語が厳密に同一の意味を持つとはいえないため，同一の概念に属する語を一律に概念に置換することで共起に関する情報が失われる可能性がある．本稿では概念への抽象化により，よりコンパクトな決定木を作成できる場合にのみ抽象化を行うことで，共起に関する情報を失うことなく，抽象化する語を選択して共起関係を抽出する．またシソーラスで定義された概念の分類が粗すぎるため，共起を抽出するには不十分である場合があるため，シソーラスの概念分類より細かい分類（細分類）を推定し，この細分類を用いた共起関係を抽出する．

また Rocchio フィードバックによる各語へのスコアを検索者の検索意図への関連度と見なし，スコアの高いものを重要な関連語として推定対象に加えることで，上記 (2) に対処する．

これら改良を加えた手法により抽出された語および概念の重要な共起を，Rocchio フィードバックによる検索結果の優先順位に反映させることで検索精度の向上を図る．

以下，2 章は Rocchio フィードバックおよび従来の共起推定手法について，3 章は本稿で提案する共起推定対象語の追加方法について述べる．4 章は OHSUMED テストコレクションを用いた実験手順について，5 章で実験結果および考察を述べる．6 章において，まとめと今後の課題について述べる．

2. 本稿で用いる用語

本稿では以下の用語を用いる．

サンプル文書 検索者が必要または不要の判定をした文書を指す．

非サンプル文書 検索対象である文書データベース中の文書のうち，サンプル文書以外の文書を指す．レレバンスフィードバックによる検索での検索対象である．

検索語 特に断らない限り検索要求文中の非不要語を指す．本稿では検索要求文の語から語幹を抽出し，そこから不要語辞書に登場する語を除去した残りを検索語としている．

類義語 概念および単語間での包含関係が明示されたシソーラスにおいて，検索語と同一の概念に属する語を指す．**図 1** にシソーラスの例を示す（なお図中#で始まる数字は概念を表す）．

関連語 検索者が必要と判定したサンプル文書中に含まれる語を指す．ただし検索要求文中の検索語は関連語に含めない．

共起 1 つの文書中に登場する，ないしは登場しない語の組合せを指す．本稿では 2 つの語の組合せに限らず，1 つ以上であれば何語の組合せでもよい．この共起を含む文書を検索する検索式は，登場する語，または登場しない語を NOT で囲んだものを AND 結合することで表現できる．たとえば語 “ATM”，“ネットワーク” が存在し，“銀行”，

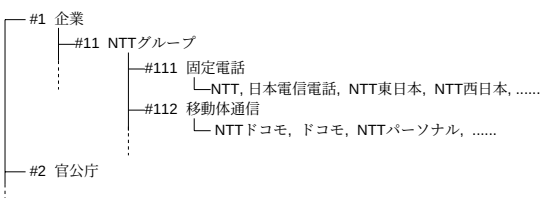

図 1 シソーラスの例
Fig. 1 An example of thesaurus.

“信用金庫” が登場しない共起を含む文書は

ATM AND ネットワーク AND NOT(銀行)
AND NOT(信用金庫)

で検索できる．本稿では，共起を含む文書を検索する検索式を，共起の 1 つの表現として扱う．

3. 従来のレレバンスフィードバック技術

本章ではレレバンスフィードバックを実現する手法である Rocchio フィードバックについて述べる．また ID3 による検索語の共起推定手法，および推定された共起関係により検索結果の優先順位を変更する手法について述べる．

3.1 Rocchio フィードバック

Rocchio フィードバック[1]はベクトル空間法と TF/IDF 法[2]を用いた文書検索システムにおいて，レレバンスフィードバックを実現する．

ベクトル空間法[2]は文書や検索要求文をベクトル空間上のベクトルとして表現する．このベクトルは文書および文中の単語が持つ重要性を示す “重み” を要素として持つ．

TF/IDF 法は，文書データベース中の少数の文書に数多く登場する語を重要な語として扱い，大きな重みを与えることで語の “重み” を決定する[2),7),8)]．文書 d_j 中の単語 t_i の重み $w_{i,j}$ は，文書 d_j 中に単語 t_i が出現する回数 $f_{i,j}$（Term Frequency，TF）および単語 t_i が出現する文書データベース中の文書数 n_i の逆数（Inverted Document Frequency，IDF）を用いて以下の式により計算される[8)]．

$$w_{i,j} = \frac{(\log(f_{i,j}) + 1.0) * log(\frac{|DB|}{n_i})}{\sqrt{\sum_{k=1}^{N} (\log(f_{k,j}) + 1.0) * log(\frac{|DB|}{n_k})^2}}$$

なお $|DB|$ は文書データベース中の文書総数である．

Rocchio フィードバック[1]はサンプル文書を用いて検索要求文のベクトルを修正することで，検索者の意図を反映したベクトルを作成する．検索要求文のベクトルを v_q，提示した文書中から検索者が選択した必要文書 num_{rel} 件の持つベクトルの和を v_{rel}，検索者が選択しなかった文書（不要文書）num_{nonrel} 件の持つベクトルの和を v_{nonrel} としたとき，新たなベクトルは

$$v = \alpha v_q + \frac{\beta v_{rel}}{num_{rel}} - \frac{\gamma v_{nonrel}}{num_{nonrel}}$$

となる（ここで α，β，γ は定数，また重みが負の値となる語は用いない）．検索要求文に対する文書のスコアは検索式のベクトルと文書のベクトルとの内積によって計算され，検索システムはスコアの高い順に文書を順位付けしてユーザに呈示する．

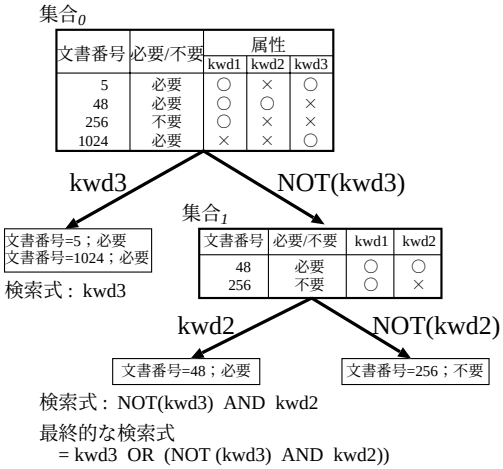

図 2 ID3 による検索式作成
Fig. 2 Producing a query using ID3.

Rocchio フィードバックで作成されるベクトル v はベクトル間の加減算によって作成されるため，検索語間の共起は反映されない．そのため検索語の重要な共起が意味を持つ文書に高いスコアが与えられない可能性がある．

3.2 決定木学習アルゴリズム ID3

ID3 は必要ないし不要の区別がされた学習例を入力として，必要例と不要例を判別する最小の決定木を獲得する機械学習アルゴリズムである[5)]．ID3 は属性の有無で学習例の集合を分割し，この分割を繰り返すことで決定木を作成する．集合の分割に用いる属性を情報量を基準として選択することで，近似的に最小の決定木を獲得する．

学習例を各文書，属性を各文書に登場する検索語とすると，決定木は検索式を木構造で表現したものと考えることができる（**図 2**）．

ID3 により作成した決定木において，必要文書を得るパスで用いた検索語を演算子 AND で結合し，各パスで得られた検索式を演算子 OR で結合したものを検索式とすると，この検索式によって検索される文書は，演算子 AND により結合された各検索語が共起する文書になる．そのため ID3 によって得られる検索式は，必要文書に存在し，不要文書には存在しない検索語の共起を表している．

ID3 およびその発展形である C4.5 は，提案者の学習例の集合を分割する際に個々の属性（本稿の例では検索語）を用いるため，複数の属性の組合せで集合分割を行う場合に比較して決定木が大きくなり，その際に未知例の判別精度が低くなりうることが提案者であ

る Quinlan 自身によって指摘されている[5]．

しかし複数のテストセットについて多くの学習アルゴリズムと比較した実験結果[9]によると，対象となった学習アルゴリズム中では比較的良好な判別精度が得られ，また学習に必要な処理時間はきわめて短い．このことから，上記の欠点は大きな欠点となることは少なく，また文書検索のように即答性が求められるシステムで用いる学習アルゴリズムとしては良好な性質を持つと考えられる．

3.2.1 Bit-per-category エンコーディング

一般に機械学習アルゴリズムで正しい決定木を学習するためには，多数の学習例が必要である．しかし実際の問題で十分な数の学習例が利用できることは少ない．この学習例の少なさを補う手段として，学習例中の属性を概念に抽象化して取り扱うことで，概念を含んだ決定木を獲得し，学習例に類似した事例を扱う手法が提案されている[10)~13)]．

Almuallim らは ID3 による決定木学習の前処理としてシソーラスを利用する Bit-per-category エンコーディングを提案している（以降この手法を **Bit** と称する）[13)~15)]．

Bit-per-category エンコーディングは各学習例について例中の語が属する概念を属性として追加し，これを ID3 に渡す．ID3 はより多くの学習例を判別できる属性を用いて学習例の集合を分割するが，この際に語を用いるより概念を用いる方がより多くの学習例を判別できる場合，概念の有無によって集合を分割する．これによって概念を含んだ検索式を得ることができる．

以下の例で Bit-per-category エンコーディングを用いた場合にどのように共起推定が行われるか示す．

必要文書 1 「NTT では IT 社会に必要な技術の開発を行っている」

必要文書 2 「日本電信電話は自社の技術を戦略の中核としている」

不要文書 1 「松芝電器は IT 技術の開発を推進している」

Bit-per-category エンコーディングは ID3 に文書を与える際に語の属する概念を新たな属性として加える（**図 3** の ‘#1’, ‘#11’, ‘#12’．図中の#で始まる数字はシソーラスの概念分類番号を表す）．

ID3 は必要文書と不要文書を最もよく判別できる属性によって学習例の集合を分割する．この場合は ‘NTT’, ‘日本電信電話’ の上位概念である ‘#11’ の有無によって最も多くの必要文書と不要文書を判別できるため，得られる検索式は ‘#11’ となる．この概念 ‘#11’ をシソーラス展開して

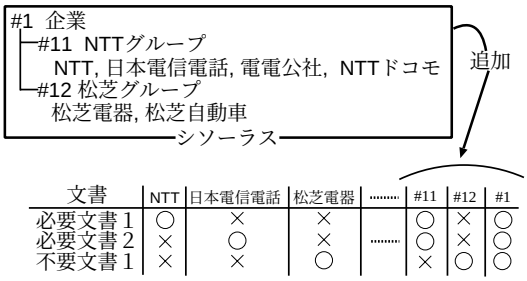

図 3 Bit-per-category エンコーディング
Fig. 3 Bit-per-category encording.

‘日本電信電話’ OR ‘NTT’ OR ‘電電公社’ OR ‘NTT ドコモ’

となることから，必要文書 1，2 に登場する語 ‘日本電信電話’ と ‘NTT’ を含んでいることが分かる．また学習例に含まれない語である ‘電電公社’ と ‘NTT ドコモ’ が検索式に含まれており，概念を用いた検索式を得ることで，検索語の類義語を含む共起が得られていることが分かる．

3.2.2 共起推定と学習例の追加

検索語の共起の中には，検索者の検索意図を表現するうえで必要不可欠なものがあり，その共起が登場する文書は他の検索語の有無に関係なく必要文書となる場合がある．Rocchio フィードバックは検索語間の共起が文書のスコアに反映されていないため，そのような重要な共起をサンプル文書から推定し，検索結果に反映できれば検索精度を向上できる．

サンプル文書集合中の必要文書にのみ登場し，不要文書には登場しない共起であっても，以下の問題点を持つ場合は非サンプル文書集合中の必要文書と不要文書の両方に登場するため，検索精度の向上には役立たない．

(1) 非サンプル文書集合中の必要文書と不要文書の両方に登場する共起の場合，このような共起を持つ文書を検索すると不要文書も検索してしまう．

(2) 非サンプル文書集合中の必要文書に登場しない共起の場合，このような共起を持つ文書を検索しても必要文書を検索できない．

より多くの検索語を含む共起ほど，上の問題点 (1) を持つ可能性が小さい．逆により少ない検索語を含む共起ほど，問題点 (2) を持つ可能性が小さい．文書データベース中の全文書について必要文書または不要文書が判定されている場合，全文書を ID3 に学習例として与えることで，問題点 (1)，(2) を持たない検索

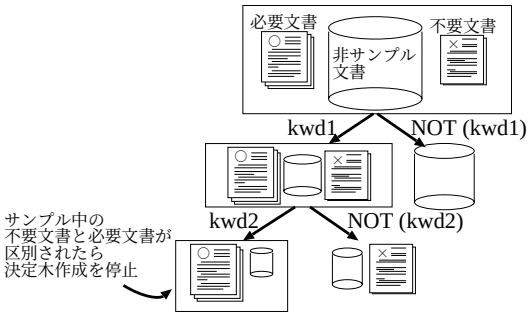

図 4 決定木作成の停止条件
Fig. 4 Stopping criteria in producing a decision tree.

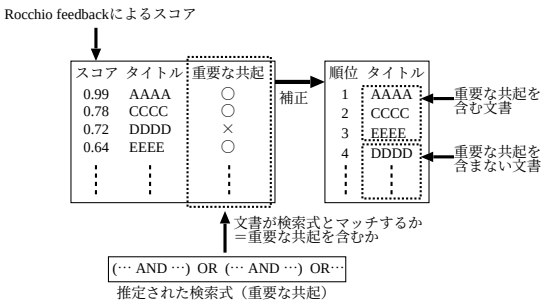

図 5 順位付けの補正
Fig. 5 Modification of ranked results.

語の共起を得ることができる．しかしサンプル文書は検索者が必要/不要の判断を下した文書であり，多くのサンプル文書が学習例として与えられることは期待できない．この場合，ID3 はより少数の単語の共起を抽出するため，抽出される共起は問題点 (2) を持つ可能性は少ないが，問題点 (1) を持つ可能性は大きい．

問題点 (1) は，ある共起が登場する不要文書が存在するにもかかわらず，その不要文書が学習例として用いられていないために発生する．このため，より多くの不要文書を学習例として与えることで問題が発生するリスクを低下させることができる．一般に文書データベースでは，検索者に必要な文書の数はデータベース全体のごく一部であり，大半の文書は不要文書と考えることができる．これを利用してすべての非サンプル文書を仮想的な不要文書として扱い，サンプル文書数の不足を補う手法が提案されている[6]．すべての非サンプル文書を不要文書として，必要文書と不要文書を判別する決定木を作成すると，サンプル文書集合中の必要文書のみを得る検索式が作成されるが，これでは検索精度の向上に役立たないため，決定木がある程度深くなったところで文書集合の分割を停止させる．本稿ではサンプル文書集合中の必要文書と不要文書が別々の集合に分かれたら，集合分割を停止させる（**図 4**）．なお以降，サンプル文書のみを ID3 に与えて共起を抽出する手法を **ID3**，上記の非サンプル文書を仮想的な不要文書として扱い，共起を抽出する手法を **Add** と称する．

3.3 順位付けの補正

検索語の重要な共起を正しく推定できれば，その共起を含む文書は含まない文書より重要と考えることができる．しかし，判定されていない文書のみに登場する共起が存在しうるため，推定された共起を含む文書のみを検索結果とすると，一部の必要文書が得られない可能性がある．Rocchio フィードバックは検索語間の共起をスコア計算に反映していないため，重要な共起を含む文書であっても与えられるスコアは必ずしも大きくならない．そのため重要な共起を含む文書が高い順位を持つとは限らないが，検索精度を向上させる手段として有効性が確認されている．このため，共起が登場する文書についてのみ Rocchio フィードバックによる順位付けを変化させることで，検索精度の向上を図る．

本稿では推定された共起を以下の手法で順位付けに反映する．

補正手法 1 推定された共起はすべて正しく，また共起を含む文書は必ず必要文書であり，含まない文書より重要な文書であるとして扱う．重要な共起を含む文書に共起を含まない文書より高い順位を与え，なおかつ重要な共起を含む文書間，および共起を含まない文書間では Rocchio フィードバックによる順位付けを維持する[6]（**図 5**）．

(1) Rocchio フィードバックにより各文書にスコアを与える．
(2) スコアを与えられた文書のうち，共起を含む文書を $d_1, d_2, \cdots, d_m$，共起を含まない文書を $d'_1, d'_2, \cdots, d'_n$ とする．
(3) $d_1, d_2, \cdots, d_m$ を Rocchio フィードバックによるスコアの高い順にソートして順位 $1, 2, \cdots, m$ 位を与える．
(4) $d'_1, d'_2, \cdots, d'_n$ を Rocchio フィードバックによるスコアの高い順にソートして順位 $m+1, m+2, \cdots, m+n$ 位を与える．

補正手法 2 推定された共起を含む文書のスコアを一律に 2 倍する．共起を含む文書の順位は上昇するが，補正手法 1 とは異なり，必ずしも共起を含まない文書より上位になるとは限らない．共起を含む文書間，また含まない文書間での順位付けは変化しない．手法 2 は共起を含む文書について Rocchio フィードバックのスコアを一律に 2 倍す

るが，手法 1 による順位補正はこの倍率を非常に高い値とした場合に相当する．

4. 共起推定対象語の追加

従来手法では検索要求文中の検索語のみを共起抽出の対象としていたため，サンプル文書に検索語が登場せず，類義語のみが登場する場合には，共起推定の対象とすることができなかった．また検索要求文中には登場せず，サンプル文書中にのみ登場する語は共起推定の対象とならないため，関連語を含む共起を推定することはできなかった．このため共起推定の対象となる語を追加することで，より多くの重要な共起関係を推定でき，検索精度を向上できると考えられる．

文書中のすべての語を学習例の属性として扱い，ID3 による共起推定の対象とすることで，文書中に登場するすべての重要な共起を推定対象とすることができる（この手法を **All** と称する）．しかし検索者が必要とする文書であっても，文書の一部分のみが検索要求と関係があり，検索要求と関係のない語が多く含まれる場合がある．このような文書中のすべての語を推定対象に含めると，学習例に不要な属性が多く含まれることになり，誤った共起が推定される可能性が高くなる．そこで本稿では

- 検索要求文中の各検索語の類義語
- 検索者に必要と判定された文書に含まれる語（関連語）

の 2 つのみを共起抽出の対象に加えることで，不要な属性の追加を避ける．

4.1 シソーラスによる共起推定対象語の追加

本節ではシソーラスを参照することで得られる類義語を共起推定の対象に加える手法について述べる．

4.1.1 Bit-per-category エンコーディングの問題点

3.2.1 項であげた Bit-per-category エンコーディングにおいて，扱う属性を検索要求文中の検索語，類義語および検索語が属する概念に限定することで，学習例中の検索語と類義語の共起を推定することが可能になる．

しかしシソーラスの分類方法によっては，特定の分類があまり細分化されておらず，1 つの分類が多くの意味を含んでいる場合がある．このような分類はより細かい分類（細分類）が存在するにもかかわらず，それら細分類がシソーラスでは省略され，その上位の分類に語がまとめられていると見なすことができる．

この細分類の省略が原因で，手法 **Bit** では以下の問題が生じうる．

(A) 特定の細分類 C についてのみ学習例が存在し，その細分類が重要な共起に含まれると推定された場合，細分類 C と同じ分類に属する他の細分類 C' も重要な共起を構成していると推定されることになる．このとき細分類 C と C' の間で指し示す意味の違いが大きい場合，細分類 C' に属する語を含む共起は誤った共起である可能性があり，検索精度を低下させる恐れがある．

(B) 細分類を共起推定の対象とすれば必要文書にのみ含まれる共起を推定できるが，実際には分類が細分化されていないために，共起推定に分類を用いても必要文書にも不要文書にも含まれる共起を推定できないことがありうる．このため重要な共起を抽出できない恐れがある．

4.1.2 細分類の推定

上記の問題点 (A)，(B) 以外にも 3.2.2 項であげた問題点 (1)，(2) を持つ共起は検索精度の向上に役立たない．

手法 **Add** は共起推定において問題点 (1)，(2) が生じるリスクを小さくする手段である．この手法 **Add** により増加させた学習例について，手法 **Bit** により属性を追加すると，検索対象となる文書データベース中で検索語と同じ概念に属する語のすべてが学習例に登場することになる．問題点 (A) は学習例に登場しない細分類を共起の一部とするため発生するが，この場合は学習例に登場していない語は検索対象となる文書データベースにも登場しないので，検索精度に悪影響を与えない．

さらに手法 **Add** と手法 **Bit** を以下のように拡張することで前記の問題点 (A)，(B) の両方を解決する手段となる（以降 **Add+** と称する）．

(1) 手法 **Add** と同様にサンプル文書と非サンプル文書からなる学習例の集合を作成する．

(2) 任意の学習例の集合 U について以下の手順で集合を分割する属性を決定し，新たな集合を作成する．

  (a) シソーラスの分類ごとに，集合 U 中の必要文書に登場する語の集合を作成する．

  (b) 分類ごとに，(a) で作成した集合に属する語を 2 語以上含むすべての組合せを作成する．組合せのうち検索語を含むものが，検索語の属する細分類である可能性があり，それらに新たな分類番号を割り当てる．

  (c) 手法 **Bit** と同様に学習例中の語が属する分類を属性として追加する．さらに語が

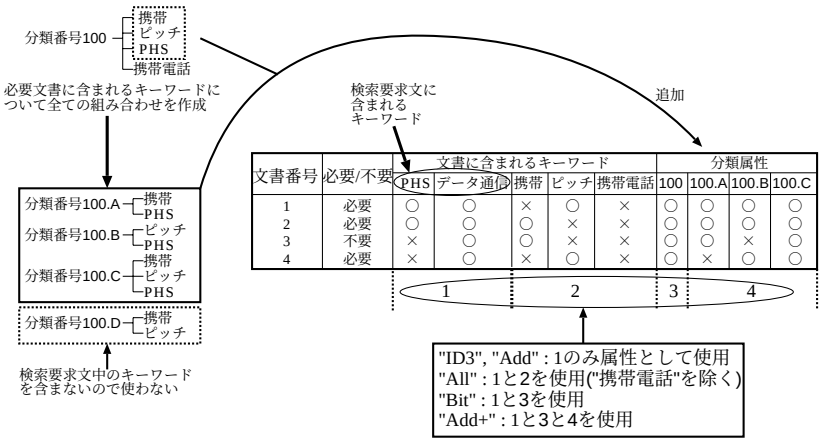

図 6 提案手法による属性追加
Fig. 6 Attributes added in our method.

(b) で分類番号が割り当てられた細分類に属する場合，その細分類も属性として追加する．

(d) ID3 と同様に情報量を基準として学習例を分割する属性を選択し，属性の有無で学習例を新たな集合 U', U'' に分割する．

(3) 手法 **Add** と同様に，いずれかの集合中に必要文書と不要文書の両方が含まれており，なおかつ分割に用いていない属性があれば (2) に戻る．なければ終了する．

検索要求文 “PHS によるデータ通信” を例に，上記手法により追加される属性を**図 6** に示す．図中では “携帯”，“ピッチ”，“PHS”，“携帯電話” が同一の分類 “100” に分類されているシソーラスを用いている．またサンプル文書 1，2 には単語 “PHS” が，文書 1，2，3，4 のすべてに “データ通信” が，文書 2，3 に “携帯” が，文書 1，4 に “ピッチ” が登場する．

図 6 では “携帯”，“ピッチ”，“PHS” が必要文書中に登場しているため，この 3 語の組合せのうち検索要求文に含まれる “PHS” を含む組合せを新たな分類 “100.A”，“100.B”，“100.C” として作成する．必要文書と不要文書の分割においては分類 “100” とともに “100.A”，“100.B”，“100.C” を属性として用いる．

問題点 (A) は学習例に登場しない細分類を共起に含めるために発生するが，**Add+**では **Add** と **Bit** を組み合わせた場合と同様に検索対象となる文書データベース中で検索語と同じ概念に属する語のすべてが学習例に登場しているため，検索精度に悪影響を与えない．

問題点 (B) は細分化されていない分類を用いるために発生するが，**Add+**では分割対象の集合において，必要文書に登場する検索語と同一概念に属する語についてすべての組合せを作成するため，必要文書に登場する細分類はすべて属性として使われることになる．このため問題点 (B) が発生する可能性は小さくなる．

4.2 関連語抽出による共起推定対象語の追加

Rocchio フィードバックにより作成されるベクトルには検索要求文に含まれる単語だけではなく，フィードバックされた文書中にのみ登場する単語もベクトルの要素として加える．ここで加えられる単語の数によって Rocchio フィードバックの精度は異なるが，一般に多くの単語を加えるほど精度は高くなる．検索要求文に登場しない単語をベクトルの要素に加えて精度が高くなるということは，加えられた単語が検索者の検索意図と非常に関連が強いものであることを示している．もし検索者が検索対象について事前に十分な知識を持っており，検索要求文により多くの単語を加えることでより正確に検索意図を入力していたとすれば，検索要求文にそれらの単語が使われていた可能性が高いと考えられる．つまり Rocchio フィードバックにより追加される単語は潜在的には検索要求文に登場していた単語であると考えられる．

従来手法では実際に検索要求文中に登場する単語のみについて共起を推定しており，これら潜在的には検索要求文に登場していた単語は推定対象としなかった．本稿では Rocchio フィードバックによるスコアを検索者の検索意図への関連度と見なし，スコアの高い単語を潜在的には検索要求文に登場していた可能性の高い関連語として，共起推定の対象語に加える．また一般に Rocchio フィードバックで加える単語の数が増加するに従い，検索精度の伸びは小さくなる．これはスコ

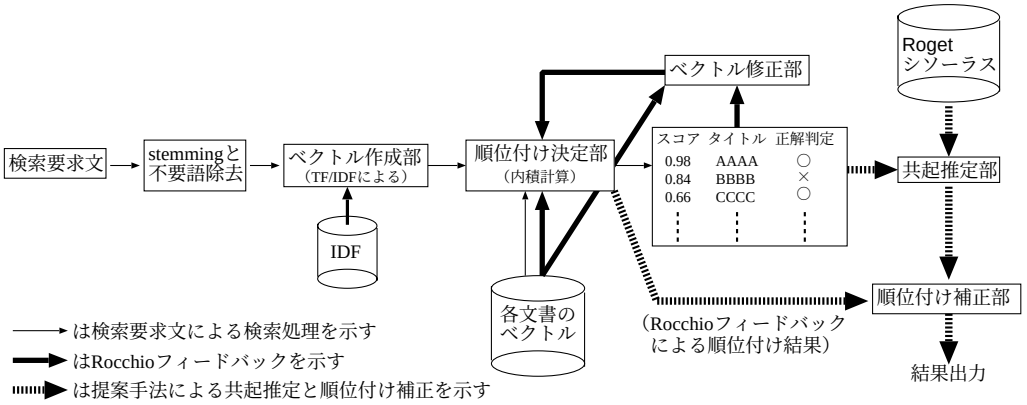

図 7 処理の流れ
Fig. 7 Processing flow.

表 1 OHSUMED テストコレクション
Table 1 Statistics of the OHSUMED test collection.

文書数	文書総量 (MB)	検索要求文数	平均質問語数
348566	381	106	6.7

表 2 Roget シソーラス
Table 2 Statistics of the Roget's thesaurus.

分類数	単語数（重複なし/重複あり）	1 分類あたり平均単語数
1041	12131/34921	33.5

アが低くなるにつれて検索意図との関連が小さくなると考えられ，共起推定においても同様の現象が起こると想定できる．これを 5 章において共起推定対象に加える関連語数を 10，20，30 語と変化させた場合の検索結果を評価することで検証する（5 章では関連語数 10，20，30 語での結果を各々**+10**，**+20**，**+30** で表す）．

5. 実　　験

本章では実験に用いたデータと実験環境，実験手順について述べる．

5.1 実験に用いたデータ

検索精度の評価には，無償で入手可能な文書検索テストコレクションとしては最も多くの文書数を持つ OHSUMED テストコレクションを用いた（**表 1**，対象文書は医学分野の文献の要約）．テストコレクションは文書の集合と検索要求文からなり，質問文に対して関連する文書（正解）が与えられている．テストコレクションの各検索要求文，文書からは FreeWAIS-sf[16] の不要語辞書に登場する語を除去後，Porter の stemming アルゴリズム[17] により語幹を取り出して利用した．なお表中の平均質問語数は各検索要求文に含まれる検索語数の平均であり，不要語除去後のものである（除去前は平均 13.4 語）．

シソーラスは Roget シソーラス[18] を用いた（**表 2**）．なお単語は OHSUMED テストコレクションに登場するもののみを用い，概念分類は各手法を比較する都合から最下層の分類（最も細かい分類）のみを用いた．なお複数の分類に属している単語があるため，表 2 には複数の分類に属する単語を一度のみカウントしたもの（“重複なし”）と複数の分類に属する単語を複数回数えた場合の単語数（“重複あり”）を示した．

5.2 実 験 手 順

実験システムの処理を**図 7** に示す．以下に実験手順を示す．

(1) 質問文から TF/IDF 法を用いて v_q を作成し，各文書のベクトルとの内積を計算して各文書のスコアとする（通常の検索，以下 **Query** と呼ぶ）．

(2) スコア上位 n $(n = 10, 20, 30)$ 件をそれぞれサンプル文書とし，テストセットの正解を用いて正解（= 必要文書）と不正解（= 不要文書）を判定する．

(3) Rocchio フィードバックにより，各文書のスコアを再計算する（以下 **Rocchio** と呼ぶ）．α，β，γ は文献 8) より各々8，16，4 とした．

(4) ID3 および **Add** に 4 章の共起推定対象語を追加する各手法を組み合わせることで重要な共起を推定する．

(5) 得られた共起を用い，3.3 節で述べた各順位補正手法により Rocchio フィードバックによる順位を補正する．

(6) 順位付けされた検索結果を適合率平均☆によっ

☆ 適合率平均は trec_eval において “Average precision (non-interpolated) over all rel docs” の項で出力される．

表 3 共起推定対象の属性と用いる学習例
Table 3 Attributes and samples for estimation.

手法	用いる属性					学習例	
	検索語	文書中のすべての語	重みの大きい語	概念	細分類	サンプル文書	非サンプル文書
ID3	○					○	
ID3,All	○	○				○	
ID3,Bit	○			○		○	
ID3,+10,+20,+30	○		○			○	
Add	○					○	○
Add,All	○	○				○	○
Add,Bit	○			○		○	○
Add+	○			○	○	○	○
Add,+10,+20,+30	○		○			○	○

て精度評価する（情報検索コンテスト TREC における評価プログラム trec_eval [19]を使用）．なお検索結果が検索要求文に対する正解記事であれば正解とし，サンプル文書は正解/不正解の判定から除去した．またサンプル文書中に必要文書が複数存在する質問のみを扱った（サンプル文書数 10，20，30 でそれぞれ 58，43，39 質問を除去）．

実験で比較する手法のうち共起推定を行うものについて，共起推定の対象となる属性および用いる学習例を**表 3** に示す．

本稿で提案する手法は，共起推定対象の対象となる語を増加させることで，検索語を組み合わせるだけでは推定できない共起を推定することを目的としている．提案手法が対象とするフィードバックと対象としないフィードバックを区別して評価するため，実験は以下の 2 種類に分別して行う．

Type A 検索語の組合せの有無では，サンプル文書中の必要文書と不要文書を判別できない場合．

Type B 検索語の組合せの有無で，サンプル文書中の必要文書と不要文書を判別できる場合．

なお検索語の組合せの有無で必要文書と不要文書が判別できるかどうか判定するために，サンプル文書のみを学習例として **ID3** で推定した検索式を用いた．得られた検索式にすべての必要文書が適合し，いずれの不要文書も適合しない場合には，検索語の組合せの有無で必要文書と不要文書が判別できるが，そうでない場合には検索語の組合せでは必要文書と不要文書を判別できない．判別できない場合を Type A に分類する．判別できる場合を Type B に分類する．

サンプル文書ごとに Type A，Type B に属するフィードバックの数を**表 4** に示す．

なお Type A，Type B の判定には検索者からフィードバックされるサンプル文書のみを用いており，非サンプル文書，つまり検索対象となる文書については判定に用いない．このため Type A，TypeB は共起推定の前に自動的に判定可能である．

表 4 Type A，B に属する質問数
Table 4 Number of queries classified into Type A or B.

種別	$n = 10$	$n = 20$	$n = 30$
Type A	25	46	58
Type B	23	17	9

表 5 適合率平均（Type A）
Table 5 Average precision (Type A).

手法	$n = 10$	$n = 20$	$n = 30$
Query	13.72	10.63	9.79
Rocchio	18.88	19.86	20.63
ID3	19.16	19.46	20.19
ID3,All	17.49	15.00	16.19
ID3,Bit	17.47	17.71	18.64
ID3,+10	17.57	18.95	19.28
ID3,+20	17.76	18.33	20.75
ID3,+30	17.18	18.44	20.65
Add	19.88	21.23	23.07
Add,All	18.20	20.52	22.58
Add,Bit	24.52	22.46	24.57
Add+	25.67	23.53	25.82
Add,+10	18.16	20.80	23.43
Add,+20	18.21	21.22	22.42
Add,+30	18.29	21.17	21.80

6. 実験結果と考察

本章では実験結果とその考察について述べる．6.1 節で 3.3 節で述べた順位補正手法 1 により順位を補正した実験の結果と考察を，6.2 節で順位補正手法 2 での実験結果と手法 1 との比較を述べる．

6.1 順位補正手法 1 による実験

従来の共起推定手法 **ID3**，**Add** と 4 章の共起推定対象語を追加する手法で推定された共起で Rocchio フィードバックの順位を補正した結果を**表 5**，**表 6** に，また上位文書の適合率を**表 7**，**表 8**，**表 9** に示す（なお表中の単位は%, n はサンプル文書数である）．順位の補正は 6.1 節の順位補正手法 1 による．表中 “**Add,All**”

表 6 適合率平均（Type B）
Table 6 Average precision (Type B).

手法	$n = 10$	$n = 20$	$n = 30$
Query	13.28	7.17	6.00
Rocchio	21.29	17.00	17.96
ID3	25.22	20.62	22.55
ID3,All	23.90	21.33	26.51
ID3,Bit	24.39	22.83	17.47
ID3,+10	26.43	22.10	19.43
ID3,+20	24.50	21.47	20.62
ID3,+30	24.35	20.92	22.20
Add	30.87	26.72	22.88
Add,All	30.01	24.39	21.43
Add,Bit	29.68	26.40	22.13
Add+	29.66	26.70	23.18
Add,+10	29.88	24.24	26.61
Add,+20	30.09	24.17	22.82
Add,+30	30.01	23.55	21.31

表 7 上位文書の適合率平均（$n = 10$，Type A）
Table 7 Average precision of top-ranked documents ($n = 10$, Type A).

手法	top10	top20	top30	top100
Query	22.80	21.60	20.53	14.96
Rocchio	32.00	27.80	23.60	18.84
ID3	34.40	30.00	25.20	18.32
Add	38.40	32.80	27.47	19.20
Add,Bit	40.00	37.00	34.67	21.40
Add+	44.00	39.40	35.20	21.72

表 8 上位文書の適合率平均（$n = 20$，Type A）
Table 8 Average precision of top-ranked documents ($n = 20$, Type A).

手法	top10	top20	top30	top100
Query	18.26	17.72	16.49	12.28
Rocchio	29.35	25.43	24.57	17.63
ID3	30.65	26.41	24.86	17.09
Add	36.52	30.11	26.88	17.78
Add,Bit	36.30	31.63	28.33	18.50
Add+	39.57	33.04	29.78	19.13

表 9 上位文書の適合率平均（$n = 30$，Type A）
Table 9 Average precision of top-ranked documents ($n = 30$, Type A).

手法	top10	top20	top30	top100
Query	15.52	14.74	14.48	10.52
Rocchio	29.31	26.64	24.77	16.28
ID3	31.55	27.07	24.54	15.53
Add	38.97	31.31	27.82	17.02
Add,Bit	38.45	31.64	28.39	18.69
Add+	39.31	32.41	29.02	19.00

など **ID3**，**Add** と共起推定対象語を追加する手法が組になっているものは，手法 **ID3**，**Add** に共起推定対象語を追加する手法を適用して共起推定を行ったことを示す．

表 10 概念属性使用の効果（Type A）
Table 10 Effect of conceptual attribute for estimation (Type A).

手法	$n = 10$	$n = 20$	$n = 30$
ID3	19.16	19.46	20.19
ID3,T1	18.20	17.04	18.54
ID3,Bit	17.47	17.71	18.64
Add	19.88	21.23	23.07
Add,T1	23.22	21.82	23.04
Add,Bit	24.52	22.46	24.57
Add+	25.67	23.53	25.82

従来の共起推定手法である **Add** は **ID3** より優れた精度を示しており，またいずれのサンプル文書数においても **Rocchio** より優れた精度を示している．本実験は文献 6) で用いられた NPL テストコレクション（質問数 93，文書数 11429，文書総量 3.1 MB）より多くの質問と文書を持つ OHSUMED テストコレクションを用いているが，NPL テストコレクションを用いた実験結果と同様に，疑似的な学習例を追加する手法 **Add** の効果が確認できる．

文書中のすべての語を共起抽出の対象となる手法 **All** と **Add**，**ID3** の組合せは，Type A，B いずれにおいても検索語のみをもちいる **Add**，**ID3** より精度が低下している．また Rocchio フィードバックによるスコアの高い語を加える手法 **+10**，**+20**，**+30** による効果は見られない．これら語のみを属性として追加する手法に対して，シソーラスを用いて概念も属性として加える手法 **Bit** と **Add** の組合せ，および **Add+** は Type A において精度向上効果を示している．

Bit，**Add+** と **All**，**+10**，**+20**，**+30** では属性として扱う語が異なるが，扱う語の違いだけが精度に影響を与えているわけではない．このことを以下の手法 '**T1**' との比較により示す．

手法 'T1' 学習例の属性として検索語の類義語を加え，概念を属性として追加せずに共起を推定するもの．'日本電信電話' と 'NTT' が同一の概念分類 '#111' に属するシソーラスを用いる場合，検索語に '日本電信電話' があれば，類義語 'NTT' を含む学習例では 'NTT' を属性として用いる．

手法 **T1** と **ID3**，**Add** を組み合わせて共起推定を行った場合の適合率平均を**表 10** に示す（表中の単位は%，n はサンプル文書数である）．

手法 **T1** は類義語が使われている文書からも重要な共起を抽出できる．しかし 1 つの検索語について複数種類の類義語が存在し，文書によって用いられる類義語が異なる場合，実際には同一の概念を指しているにもかかわらず，個々の類義語が集合分割に用いる属

性選択の対象となる．この場合，個々の類義語が登場するか否かでは多くの必要文書と不要文書を判別できないため，他のあまり重要でない語の有無で必要文書と不要文書が判別できると，類義語が集合分割に用いられず，重要な共起の一部として抽出されない恐れがある．

これに対して手法 **Bit** では語よりも概念で集合分割を行う方がより多くの必要文書と不要文書を判別できる場合に，検索語とその類義語を概念を表す 1 つの属性で扱い，概念の有無によって集合が分割される．この場合，検索語とその類義語が同一の属性として扱われるため，学習例中での登場回数は増加し，手法 **T1** よりノイズの影響を受けにくい．**T1** と **Bit** の違いは **Bit** が概念を属性として用いている点のみであり，精度の違いは概念を集合分割に用いたために生じている．

手法 **+10**，**+20**，**+30** は語のみを属性として追加する点では **T1** と同じであるが，**T1** ほどの効果が図れないことから，検索語として精度向上に役立つ語であっても，共起推定の対象としては必ずしも有用でないことが分かる．

手法 **Bit** は **ID3** と組み合わせた場合には **ID3** より精度が悪化する場合があるが，**Add** と組み合わせた場合にはいずれのサンプル文書数でも精度が向上している．**Bit** は問題点 (A)，(B) を持つが，このうち **Add** と組み合わせた場合には問題点 (A) が悪影響を与えない．**Bit** と **Add** の組合せが精度を向上させているのに対して，**Bit** と **ID3** の組合せが精度を悪化させているのは，この問題点 (A) が影響していると考えられる．

Add+はいずれのサンプル文書数でも最良の検索精度となっており，また **Add** と **Bit** の組合せより高い精度を示している．**Add+**は細分化された分類を属性として追加することで問題点 (B) を避けており，この効果が現れていると考えられる．

また Type B においては，いずれの共起推定対象を追加する手法も従来手法を上回る効果が得られない．検索語の組合せの有無のみで必要文書と不要文書を区別できる場合には，本稿で提案する共起推定対象語を追加する手法は効果がなく，検索語の組合せの有無だけでは不十分な場合に本稿の提案手法が効果を示すといえる．

6.2 順位補正手法 2 による実験

推定された共起を用いて順位補正手法 2 により順位を変化させた結果を**表 11** に示す．

また**表 12**，**表 13** に補正手法 1，2 のそれぞれについて各質問ごとの適合率平均を Rocchio フィードバックと比較した結果を示す．表の比較の欄は順位補正後の検索精度（適合率平均）が Rocchio フィードバックより向上したか悪化したかを示す．X/Y の X は適合率平均の差分を平均したもの，Y は該当する質問数を示す．たとえば表 12 の “手法 = **Add**，比較 = 向上，$n = 10$” の欄は，**Add** は $n = 10$ とした実験において Rocchio フィードバックより適合率平均が向上した質問が 7 件あり，適合率平均は 7 質問の平均で 10.76%上昇したことを示す．

なお本節では補正手法 1 が最も効果を示した条件である，手法 **Add** を Type A の質問に対して適用した結果のみを示す．

補正手法 2（表 11）は補正手法 1（表 5）と同程度の精度向上効果を示している．表 12，表 13 の比較から補正手法 1 は手法 2 に比べて精度が悪化する質問数，精度の変動ともに大きいことが分かる．手法 1 では Rocchio フィードバックによるスコアがきわめて小さい文書であっても，共起を含む場合には，いずれの共起を含まない文書よりも高い順位が与えられるが，手法 2 ではスコアは上昇するものの，共起を含まない文書より高い順位になるとは限らない．特に Rocchio フィードバックによるスコアの小さい文書のスコアを

表 11 適合率平均（Type A，順位補正手法 2）
Table 11 Average precision (Type A, revision method 2).

手法	$n = 10$	$n = 20$	$n = 30$
Add	20.21	21.86	23.65
Add,Bit	24.89	23.39	24.90
Add+	25.87	23.86	25.50

表 12 Rocchio フィードバックとの精度比較（補正手法 1）
Table 12 Improvement compared to the Rocchio feedback (revision method 1).

手法	比較	$n = 10$	$n = 20$	$n = 30$
Add	向上	10.76/7	7.82/19	12.04/19
	悪化	16.73/3	9.07/8	5.60/17
Add,Bit	向上	14.20/12	9.82/20	10.25/28
	悪化	3.88/8	3.58/16	3.77/19
Add+	向上	14.82/14	10.06/20	11.87/29
	悪化	10.23/4	3.01/15	5.19/17

表 13 Rocchio フィードバックとの精度比較（補正手法 2）
Table 13 Improvement compared to the Rocchio feedback (revision method 2).

手法	比較	$n = 10$	$n = 20$	$n = 30$
Add	向上	10.31/7	7.19/21	10.04/23
	悪化	12.93/3	9.76/6	4.19/13
Add,Bit	向上	12.90/13	8.42/24	10.06/29
	悪化	2.45/7	3.28/12	2.37/18
Add+	向上	14.43/14	8.61/25	11.28/31
	悪化	6.81/4	3.06/10	4.39/15

2倍しても，手法1ほど順位が上昇することはない．このため推定された共起に誤りがある場合，手法1の方が手法2よりも大きく精度が悪化すると考えられる．

7. おわりに

レレバンスフィードバックにおいて，従来は検索要求文中の検索語についてのみ推定していた重要な共起関係を，シソーラスを参照することで検索語の類義語を共起推定対象に加える手法と，Rocchioフィードバックにより得られる重みの大きい語（関連語）を推定対象に加える手法を提案し，実験によりその精度向上効果を検証した．各種の共起推定手法のうち，シソーラスを参照して語の属する概念分類を共起推定対象とするとともに，より細分化された分類も共起推定に加える手法 **Add+**が最も優れた精度向上効果を示した．

細分類を推定する手法 **Add+**では，同一の概念に属する語の組合せのそれぞれを細分類とするため，必要文書中の多くの語が同一の分類に属する場合には多数の細分類が作成され，処理に時間がかかる☆．文書検索システムのインタフェースとして実用的な処理時間を実現するためには，組み合わせる語の数に制限を加えるなど，本稿で用いた手法を近似する方法で処理時間の短縮を図る必要がある．

本稿ではRogetのシソーラスを実験に用いたが，シソーラスにはより細かく語を分類したものもある．語がより細かく分類されたシソーラスは4.1.1項で述べた細分類の省略があまり行われていないと考えられる．しかし，その分類が検索者の意図と一致していない場合，分類を用いても検索精度を向上することはできない．一方，あまり細かく語が分類されていないシソーラスは，利用者の視点によって多様な分類が可能であるため，手法 **Add+**で検索要求ごとに適切な細分類を推定できれば，検索者の多様な検索意図に対応できると考えられる．検索対象となる文書の分野や検索要求文の種類によって，どのようなシソーラスを利用するのが適当か評価する必要がある．

3.2節でID3の欠点として属性の組合せを学習例の分割に用いないために学習結果が悪くなる可能性が指摘されていることを述べたが，この問題を解決するために，属性の組合せを選択的に作成し，これを学習例の集合の分割に用いることで，より精度の高い学習結果を得る手法が提案されている[20]．この手法の処理時間をより削減して適用する手法を検討中である．

順位補正手法2は共起を含む文書についてRocchioフィードバックのスコアを一律に2倍しているが，この倍率を変化させることで精度向上の効果が変化する（たとえば手法1による順位補正はこの倍率を非常に高い値とした場合に相当）．推定される共起がサンプル文書集合中の必要文書の多くに含まれているなど，共起が正しく推定されたと考えられる場合には高い倍率でスコアを増加させ，少数の必要文書にしか含まれない場合には倍率を低くすることで，より精度を向上できると予想できる．

また順位補正手法1，2では推定された共起が各文書でどの程度重要であるのか考慮していないが，ベクトル空間法の枠組みの中で検索式（共起）と文書の内積を計算するモデルがこれまでにいくつか提案されており[21]，これを用いることで推定された共起が文書中でどの程度の重みを持つかによって，異なる扱いをすることができる．Fuzzyセットモデル[22),23)]やP-normモデル[21),24),25)]により検索式（共起）と文書の内積を計算し，これをスコアに加算する手法を検討中である．

☆ 本稿で述べた実験では共起推定処理に1質問で160分程度必要とする場合がある（使用機材はCPUにIntel Celeron 300A (450 MHz)，メモリ512 MB，OSはlinux 2.0.35）．

参考文献

1) Rocchio, J.J.: Relevance feedback in information retrieval, *The SMART Retrieval System*, pp.313–323, Prentice-Hall (1971).
2) Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill Advanced Computer Science Series, McGraw-Hill Publishing Company (1983).
3) Harman, D.: Overview of the Second Text REtrieval Conference (TREC2), *2nd Text REtrieval Conference* (*TREC-2*), pp.1–20, Department of Commerce, National Institute of Standards and Technology (1994).
4) Harman, D.: Overview of the Second Text REtrieval Conference (TREC3), *3rd Text REtrieval Conference* (*TREC-3*), pp.1–20, Department of Commerce, National Institute of Standards and Technology (1995).
5) Quinlan, J.R.: *C4.5: Programs for machine learning*, Morgan Kaufmann (1993).
6) 中島浩之，木谷　強，岡田　守：検索語間における共起関係の特定によるレレバンスフィードバックの高精度化，情報処理学会論文誌，Vol.40, No.3, pp.1236–1244 (1999).
7) 海野　敏：出現頻度情報に基づく単語重みづけの原理，*Library and Information Science*, pp.67–87 (1988).
8) Buckley, C., Salton, G. and Allan, J.: Using Query Zoning and Correlation With in SMART: TREC5, *TREC-5* (1997).

9) Lim, T.-S., Loh, W.-Y. and Shin, Y.-S.: A Comparison of Prediction Accuracy, Complexity and Training Time of Thirty-Three Old and New Classification Algorithms, *Machine Learning*, Vol.40, pp.203–228 (2000).
10) Haussler, D.: Quantifying inductive bias: AI learning algorithms and Valiant's learning framework, *Artificial Intelligence*, Vol.26, No.2, pp.177–211 (1988).
11) Michalski, R.S.: A Theory and Methodology of Inductive Learning, *Artificial Intelligence*, Vol.20, pp.111–116 (1983).
12) Nunez, M.: The Use of Background Knowledge in Decision Tree Induction, *Machine Learning*, Vol.6, pp.231–250 (1991).
13) フセイン・アルモアリム，秋葉泰弘，金田重郎：木構造属性を許容する決定木学習，人工知能学会誌，Vol.12, No.3, 人工知能学会 (1997).
14) Almuallim, H.: Two Methods for Learning ALT-J/E Translation Rules from Example and a Semantic Hierarchy, *Proc. COLING94*, pp.57–63 (1994).
15) Almuallim, H., Akiba, Y. and Yamazaki, T.: Induction of Japanese-English Translation Rules From Ambiguous Examples and a Large Semantic Hierarchy, 人工知能学会誌，人工知能学会 (1994).
16) Pfeifer, U. and Huynh, T.: FreeWAIS-sf (1994).
ftp://ls6-www.infomatik.uni-dortmund.de/pub/wais/freeWAIS-sf-1.0.tgz
17) Porter, M.F.: An Algorithm For Suffix Stripping, *Journal of the Society for Information Science*, Vol.14, No.3, pp.130–137 (1980).
18) Roget, P.M.: *Thesaurus of English words and phrases* (1852).
19) Salton, G. and Buckley, C.: trec_eval.
ftp.cs.cornell.edu/pub/smart.
20) Brodley, C.E. and Utgoff, P.E.: Multivariate Decision Trees, *Machine Learning*, Vol.19, pp.45–77 (1995).
21) Lee, J.H.: Properties of Extended Boolean Models in Information Retrieval, *SIGIR*, pp.182–190 (1994).
22) Buell, D.A.: A general model of query processing in information retrieval system, *Information Processing & Management*, Vol.17, No.5, pp.249–262 (1981).
23) Sachs W.M.: An approach to associative retrieval through the theory of fuzzy sets, *Journal of American Society for Information Science*, Vol.27, pp.85–87 (1976).
24) Wu, H.: On query formulation in information retrieval, Ph.D. Thesis, Cornell University (1981).
25) Salton, G., Fox, E.A. and Wu, H.: Extented Boolean Information Retrieval, *Comm. ACM*, Vol.26, No.12, pp.1022–1036 (1983).

(平成 13 年 3 月 26 日受付)
(平成 14 年 2 月 13 日採録)

中島 浩之（正会員）

1994 年東京工業大学大学院理工学研究科情報工学専攻修了．1994 年 4 月 NTT データ通信（株）（現（株）NTT データ）入社．2000 年 5 月より NTT コミュニケーション科学基礎研究所に勤務．情報検索，テキスト処理に関する技術開発に従事．人工知能学会会員．