

# 検索語間における共起関係の特定による レレバансフィードバックの高精度化

中島 浩之<sup>†</sup> 木谷 強<sup>†</sup> 岡田 守<sup>††</sup>

レレバансフィードバックを実現する代表的な手法である Rocchio フィードバックは、検索要求文と文書をベクトルで表現するベクトル空間法において、検索者によるフィードバック情報を用いて検索要求文から作成した検索ベクトルを修正する。この手法を用い、多くの研究者が文書検索の精度を向上させる効果を報告しているが、ベクトルの修正はベクトル間の加減算によってのみ行われるため、検索語間の共起関係をとらえることができなかった。本稿では検索語の重要な共起関係を決定木学習アルゴリズム ID3 を用いて推定し、推定した共起を含む文書について Rocchio フィードバックによる順位付けを補正する手法を提案する。学習例数が不足する場合は共起関係を正確に学習できないという ID3 の欠点を補うため、文書データベース中の大部分の文書は検索者にとって関心がない文書であることに着目し、仮想的に負例を増加させる。実験の結果、提案手法により検索精度を Rocchio フィードバックに対して 5% 程度向上できることが分かった。

## Improving Relevance Feedback through Recognizing Co-occurrences of Query Words

HIROYUKI NAKAJIMA,<sup>†</sup> TSUYOSHI KITANI<sup>†</sup> and MAMORU OKADA<sup>††</sup>

Rocchio feedback is a method to modify queries based on evaluated relevant and irrelevant documents by a user. Although the Rocchio feedback is known to be effective in improving retrieval accuracy, it doesn't capture any co-occurrences of query words. We propose to use ID3 inductive learning algorithm for capturing co-occurrences of query words in order to correct the order of ranked results from the Rocchio feedback. In spite of the fact that the ID3 requires a lot of sample documents for learning proper co-occurrences, the user often does not give enough samples. Since almost all of documents in a database are irrelevant, we add non-sample documents as provisional irrelevant samples. Experimental results show that the proposed method improves the retrieval accuracy by about 5 percent compared to the Rocchio feedback.

### 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ためには、検索式を適切に作成する必要がある。しかし検索式の作成は、検索業務を専門とするサーチャですら試行錯誤を必要とする困難な作業である<sup>20)</sup>。レレバансフィードバックは検索結果の中から検索者が選択した必要文書を利用して検索式を修正し、新たな検索式を作成する手法である。これはシステムと検索者が協調して検索式を作成するものであり、文書検索精度を向

上させる有効な手段と考えられている<sup>13)</sup>。

レレバанс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<sup>12)</sup>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<sup>5)</sup> (Vector Space Model) において、検索者によるフィードバック情報を用いて文書検索の精度を向上させる手法である。

Rocchio フィードバックはベクトル空間法によるスコア付けを検索者によるフィードバック情報を用いて向上させる手法である。検索者はベクトル空間法により呈示された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検索式に追加して再度検索を行う。次に検索要求文を表すベクトルを修正し、検索された文書が持つベクトルとの内積をスコアとして付与し、検索結果をスコアの高い順に順位

<sup>†</sup> 株式会社 NTT データ情報科学研究所

Laboratory for Information Technology, NTT Data Corporation

<sup>††</sup> 高知工科大学情報システム工学部

Department of Information System Engineering, Kochi University of Technology

付けして表示する。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るべき、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知られており、多くの研究者からその有効性が報告されている<sup>7),8),18)</sup>。しかしベクトルの修正はベクトル間の加減算によってのみ行われ、検索語間の関係は考慮されない。そのため、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場合（たとえば検索要求文「マシンがクラッシュした事例」）でも、一部の検索語（たとえば「マシン」）のみが数多く登場する文書があれば高いスコアが与えられるのに対し、検索語間の重要な共起が現れる文書に高いスコアが与えられるとは限らなかった。これは Rocchio フィードバックが検索語間の共起をスコア計算に反映していないことが原因であった。

本稿ではフィードバック情報から決定木学習アルゴリズム ID3 を用いて検索語間の重要な共起を推定し、共起を含む文書の順位を上昇させることで、Rocchio フィードバックの持つ欠点を補う手法を提案する。

## 2. 既存技術

本章ではレレバンスフィードバックを実現する既存の技術である Rocchio フィードバックおよび改良 Rocchio フィードバックについて述べる。また決定木学習アルゴリズムの1つである ID3 について述べる。

なお以降、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。検索者からシステムにフィードバックされる文書であり、レレバンスフィードバック実行時には検索者にとって既知の文書である。またすべてのサンプル文書の集合をサンプル文書集合と呼ぶ。

**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の目的は非サンプル文書の集合から必要文書を検索することである。また文書データベース中のすべての非サンプル文書からなる集合を非サンプル文書集合と呼ぶ。

### 2.1 Rocchio フィードバック

Rocchio フィードバックはベクトル空間法と TF/IDF 法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<sup>5)</sup>。このベクトル空間は扱う単語の種類と等しい数の次元を持ち、文書は文書中の単語が持つ重要性を示す“重み”を要素としたベクトルによって表される。

TF/IDF 法は、文書データベース中の多くの文書に登場する語は重要でなく、特定の文書において多く登場する語は重要とすることで単語の“重み”を決定する手法である<sup>4),5),19)</sup>。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  $t_i$  が出現するデータベース中の文書数  $n_i$  の逆数 (Inverted Document Frequency, IDF) を用い、以下の式により計算される<sup>4)</sup>。

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

なお  $|DB|$  は文書データベース中の文書総数である。

Rocchio フィードバックは検索者が必要または不要の判定をした文書（サンプル文書）のベクトルを用いて検索要求文のベクトルを修正することで、検索者の意図を検索式に反映する。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）のうち、選んだ文書より上位にある文書  $num_{nonrel}$  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。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表示する（図1）。

Rocchio フィードバックで作られるベクトル  $v$  はベクトル間の加減算により作成されるため、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を持つ文書に高いスコアが与えられない可能性がある。

### 2.2 改良 Rocchio フィードバック

Rocchio フィードバックはサンプル文書が多いほど通常のベクトル空間法を用いる検索に比べて検索精度が向上する<sup>4)</sup>。しかし検索者に対して大量の文書を必要ないし不要と判断するように要求することは現実的ではない。

Buckley らは最初の検索要求文により検索された文書のうち、検索者が必要または不要のチェックをしていない文書（非サンプル文書）をすべて不要文書と仮定し、フィードバックされる不要文書の数を増加させる手

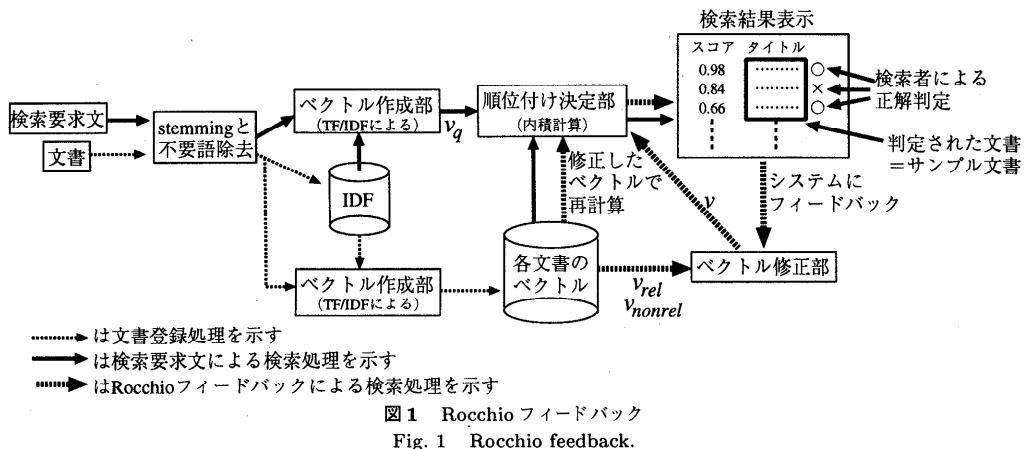


図1 Rocchio フィードバック  
Fig. 1 Rocchio feedback.

法（改良 Rocchio フィードバック, modified Rocchio feedback）を提案している<sup>3)</sup>。検索者が必要または不要のチェックをしていない文書から検索者に必要な文書を選択することがレレバансフィードバックの目的であり、それらすべてを不要文書とするのは正しい仮定ではない。しかし一般に、検索対象のデータベースにおいて必要文書が占める割合はきわめて小さく、必要文書を不要文書として扱うことによる悪影響より、不要文書を増加させることによる効果の方が大きいと報告されている。

### 2.3 決定木学習アルゴリズム ID3

ID3 は相互情報量を尺度として用いることで（近似的に）最小の決定木を作成するアルゴリズムである<sup>10)</sup>。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、ID3 により得られた決定木は容易に検索式へ変換することができる。

ID3 のアルゴリズムを以下に示す。

- (1) 入力された必要文書と不要文書の番号からなる集合を  $Set_0$  とする。
- (2) 集合  $Set_0$  に“未分割”の印をつける。
- (3) “未分割”の印がついた集合のうち任意の集合  $Set_i$  中の必要文書、不要文書中の自立語  $t_j (1 \leq j \leq N)$  について、以下の式によって相互情報量  $I(t_j)$  を計算する（“未分割”の集合がなければ終了）。

$$I(t_j) = H - H(t_j)$$

ここで

$$p_i = Set_i \text{ 中の必要文書数}$$

$$n_i = Set_i \text{ 中の不要文書数}$$

\* 最小の決定木を作成するのは MDL 原理<sup>11)</sup>による。また最小決定木の決定問題は NP 完全であり<sup>10)</sup>、ID3 のアルゴリズムは近似解を求めるものである。

$$\begin{aligned} s_i &= p_i + n_i \\ p_i(t_j) &= Set_i \text{ で } t_j \text{ を含む必要文書数} \\ n_i(t_j) &= Set_i \text{ で } t_j \text{ を含む不要文書数} \\ s_i(t_j) &= p_i(t_j) + n_i(t_j) \\ p_i(\bar{t}_j) &= Set_i \text{ で } t_j \text{ を含まない必要文書数} \\ n_i(\bar{t}_j) &= Set_i \text{ で } t_j \text{ を含まない不要文書数} \\ s_i(\bar{t}_j) &= p_i(\bar{t}_j) + n_i(\bar{t}_j) \\ h(a, b, c) &= -\left\{\frac{a}{c} \log_2\left(\frac{a}{c}\right) + \frac{b}{c} \log_2\left(\frac{b}{c}\right)\right\} \end{aligned}$$

とし、 $H$  と  $H(t_j)$  は

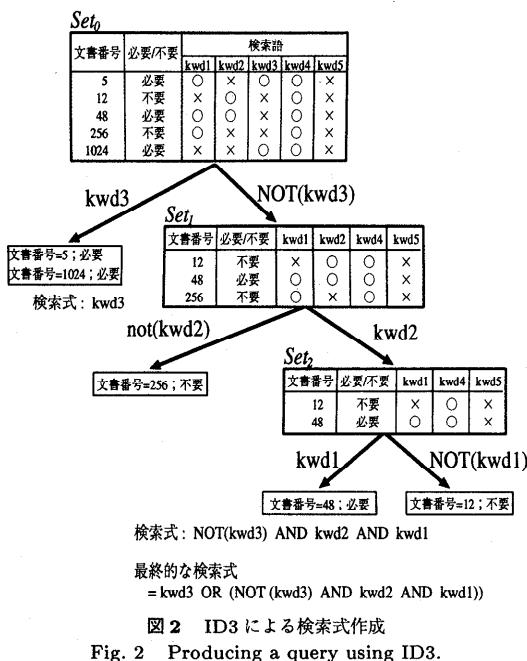
$$\begin{aligned} H &= h(p_i, n_i, s_i) \\ H(t_j) &= \frac{s_i(t_j)}{s_i} h(p_i(t_j), n_i(t_j), s_i(t_j)) \\ &\quad + \frac{s_i(\bar{t}_j)}{s_i} h(p_i(\bar{t}_j), n_i(\bar{t}_j), s_i(\bar{t}_j)) \end{aligned}$$

とする。

- (4) 自立語  $t_j (1 \leq j \leq N)$  から  $I(t_k)$  を最大にする  $t_k$  を選ぶ（複数ある場合は任意の 1 つ）。  
 $I(t_k) > 0$  の場合、 $t_k$  を持つ文書からなる集合を  $Set_{i'}$ 、持たない文書からなる集合を  $Set_{i''}$  とし、それぞれに“未分割”的印をつける。 $i'$ ,  $i''$  はすでに集合  $Set_{i'}, Set_{i''}$  が存在しなければ任意の数でよい。 $I(t_k) = 0$  の場合は分割しない。
- (5) 集合  $Set_i$  から“未分割”的印を除き、(3)へ戻る。

上記アルゴリズムで作成した決定木において、必要文書を得るパスで用いた単語を演算子 AND で結合して検索式を作成する。さらに各パスで得られた検索式を演算子 OR で結合したものを最終的な検索式とする（図 2）。

作成される検索式によって検索される文書は、AND



により結合された各単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単語の共起を表していると考えることができる。

### 3. 重要な共起による Rocchio フィードバックの出力順位の補正

本章では検索語の重要な共起を推定する手法と、推定した共起により Rocchio フィードバックによる順位を補正する手法について述べる。

#### 3.1 共起の推定

検索要求文中の検索語の組合せの中には、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、その組合せ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。本稿は検索者が判断した必要文書と不要文書を比較することで、このような検索語の組合せを重要な共起として抽出する。

重要な共起は、サンプル文書集合中の必要文書にのみ登場する（不要文書には登場しない）単語の組合せに含まれる可能性がある。この条件を満たす検索語の組合せのうち、以下の問題点を持つ組合せは非サンプル文書集合中の必要文書と不要文書を区別できない。

- (1) 必要な検索語を含まない組合せ：非サンプル文書集合中の必要文書と不要文書を区別するために必要な検索語を含まない場合、不要文書にも登場する共起となる。このような共起を持つ文

文書番号	必要/不要	検索語				
		kw <sub>d1</sub>	kw <sub>d2</sub>	kw <sub>d3</sub>	kw <sub>d4</sub>	kw <sub>d5</sub>
5	必要	○	×	×	○	×
12	不要	×	○	×	○	×
48	必要	○	○	○	○	○
256	不要	○	○	×	○	×
1024	必要	×	×	○	○	×

(1) 必要文書に登場する組合せを抽出

(kw<sub>d1</sub> AND kw<sub>d4</sub>) OR  
(kw<sub>d1</sub> AND kw<sub>d2</sub> AND kw<sub>d4</sub>) OR  
(kw<sub>d3</sub> AND kw<sub>d4</sub>)

(2) (kw<sub>d1</sub> AND kw<sub>d2</sub> AND kw<sub>d4</sub>) は不要文書にも登場するので除去

→ (kw<sub>d1</sub> AND kw<sub>d4</sub>) OR  
(kw<sub>d3</sub> AND kw<sub>d4</sub>)

図 3 手法 Combine による検索式作成

Fig. 3 Producing a query using the method "Combine".

書を検索すると不要文書も検索してしまう。

- (2) 不要な検索語を含む組合せ：非サンプル文書集合中の必要文書に登場しない検索語を用いる場合、必要文書に登場しない共起となる。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組合せほど、問題点(1)を持つ可能性が小さい。サンプル文書集合中の必要文書にのみ登場する組合せのうち、可能な限り多くの検索語を含む組合せを以下の手順で得ることができる

- (1) サンプル文書集合中の各必要文書について、文書中に登場する全検索語の組合せを抽出
- (2) 抽出した組合せから不要文書に登場する組合せを除去

この手順で得た検索語の組合せを重要な共起として扱い、共起を含む文書を検索する場合、不要文書を検索してしまう可能性は小さい（この手法を Combine と略す。図 3）。しかし、多くの検索語を用いるため、問題点(2)を持つ可能性が大きく、非サンプル文書集合から必要文書を検索できない恐れがある。

文書データベース中の全文書について必要文書と不要文書が判定されている場合、ID3 で集合分割に用いる自立語を検索語に限定して決定木を作成することで、問題点(1)と(2)を持たない検索語の共起を得ることができる。しかし通常与えられるサンプル文書集合は文書データベースの一部分であるため、ID3 によってサンプル文書集合中の必要文書と不要文書を区別する共起を推定する場合（以降、この手法を ID3 と略す）、非サンプル文書集合中の不要文書に存在する共起となる可能性がある（上記の問題点(1)に該当）。

本稿では改良 Rocchio フィードバックと同様にすべての非サンプル文書を不要文書と仮定し、ID3 に与え

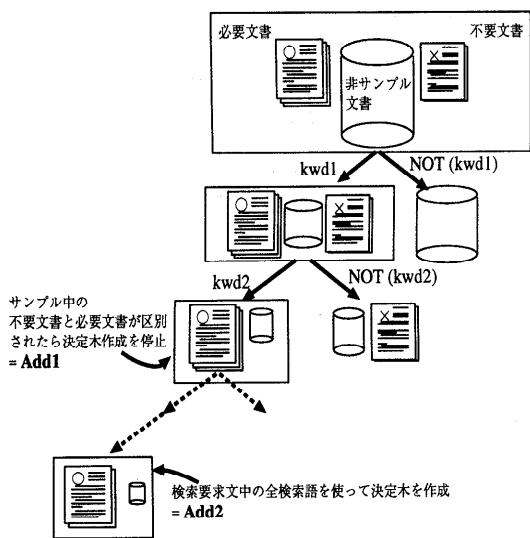


図 4 決定木作成の停止条件

Fig. 4 Stopping criterions in producing a decision tree.

る学習例を増加させることで、サンプル文書数の不足を補う。すべての非サンプル文書を不要文書とするため、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。ここでは決定木の深さがある程度深くなつたところで文書集合の分割を停止させ、その段階で必要文書を得るパスを検索式作成に用いる。

本稿では停止条件として以下の 2 種類を用いる(図 4)。なお、集合分割に用いる語は ID3 と同様に検索要求文中の検索語に限定する。

- (1) サンプル文書集合中の必要文書と不要文書が区別できたら停止する(以降 Add1 と略)。
- (2) Add1 で作成した決定木で必要文書と非サンプル文書が区別されていない集合があれば分割を継続する。いずれの検索語でも必要文書と非サンプル文書を区別できなければ分割を停止する(以降 Add2 と略。なお必要文書にも非サンプル文書にも登場しない検索語は集合分割には用いない)。Add2 で生成される決定木の葉で必要文書と同一の集合に属する非サンプル文書は、検索要求文中の検索語を組み合わせた検索式では必要文書と区別できない文書である。

Add1, Add2 の両手法とも、必要文書を含む集合を得るパスで用いた検索語を演算子 AND で結合して検索式を作成する。さらに各パスで得られた検索式を演算子 OR で結合したもの最終的な検索式とする。

### 3.2 順位付けの補正

検索者は文書検索システムが検索結果として表示す

る複数の文書を閲覧することで必要な文書を得る。一般に検索結果として得られる文書数は膨大であるため、システムは各々の文書を順位付けし、順位の高い文書から検索結果として表示する。この際にシステムがより重要性が高いと推定される文書を上位に順位付けすることで、検索者は少数の文書を閲覧するだけで必要な文書を得ることができる。

検索語の重要な共起が正しく推定でき、また非サンプル文書集合中の必要文書が必ず重要な共起を含むと仮定すると、重要な共起が登場する文書だけを検索結果とすればすべての必要文書が得られる。しかし実際には重要な共起を含まない文書の中にも必要文書が存在し、またサンプル文書に登場しない重要な共起を前節の手法によって推定することは不可能である。そのため推定した共起を含む文書のみを検索結果とすると、一部の必要文書しか得られない可能性がある。

本稿では推定した重要な共起を順位付けの補正に用いる。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。しかし Rocchio フィードバックは検索精度を向上させる手段として有効性が確認されており、共起の扱いを除けば妥当な順位付けが行われているといえる。本稿では重要な共起を含む文書の集合と含まない文書の集合に分割した場合、各々の集合内では適当な順位付けがなされていると仮定する。

前節の手法で検索語の重要な共起が正しく推定されているとすると、重要な共起を含む文書は共起を含まない文書より高い順位を持つ必要がある。本稿では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ため、以下の手順で文書を順位付けする(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  $d_1, d_2, \dots, d_m$ 、共起を含まない文書を  $d'_1, d'_2, \dots, d'_n$  とする。
- (3)  $d_1, d_2, \dots, d_m$  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  $1, 2, \dots, m$  位を与える。
- (4)  $d'_1, d'_2, \dots, d'_n$  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  $m+1, m+2, \dots, m+n$  位を与える。

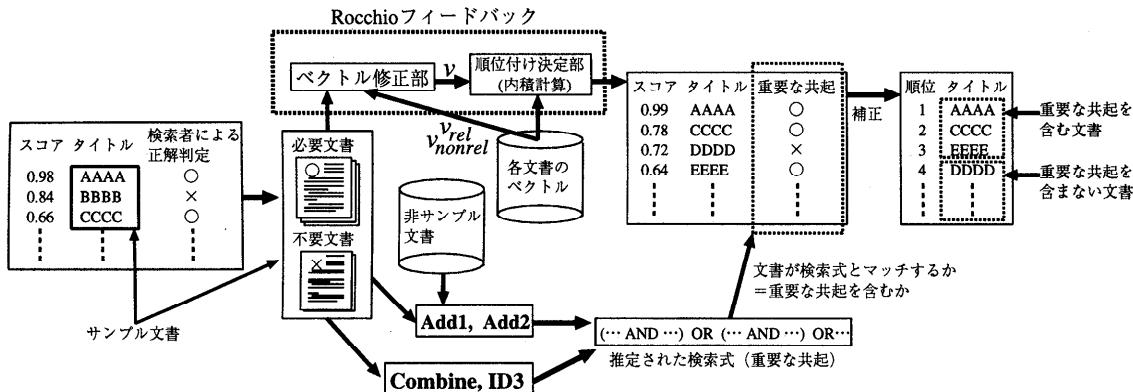


図 5 順位付けの補正  
Fig. 5 Modification of ranked results.

表 1 NPL テストコレクション

Table 1 Statistics of the NPL test collection.

文書数	文書総量 (MB)	質問数	平均質問語数	平均正解数
11429	3.1	93	6.7	22.4

## 4. 実験

本章では実験に用いたデータと実験手順について述べる。

### 4.1 使用データ

検索精度の評価には、英文を対象とした文書検索テストコレクションのうち、多くの研究者が精度評価に用いている NPL テストコレクション<sup>1),2)</sup>を用いた(表 1、対象文書は物理分野の文献の要約)。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、各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からは FreeWAIS-sf<sup>14)</sup>の不要語辞書に登場する語を除去し、さらに残った単語から Porter の stemming アルゴリズム<sup>9)</sup>により語幹のみを取り出して利用した。実験に用いたシステムの処理の流れを図 6 に示す。

### 4.2 実験手順

実験手順を以下に示す。

- (1) 検索要求文から 2 章で述べた TF/IDF 法を用いて  $v_q$  を作成、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)。
- (2) スコア上位  $n(10, 30, 50)$  件をサンプル文書とし、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。 $\alpha, \beta, \gamma$  は文献 1) より各々 8, 16, 4 とした。

- (4) サンプル文書を用いて検索要求文中の検索語について重要な共起を推定する。推定には 3.1 節で示した Combine, ID3, Add1, Add2 を用いる。
- (5) 得られた共起を 3.2 節で示した手法により Rocchio フィードバックによる順位を補正する。また共起を推定する手法と比較するため、
  - 検索要求文中のすべての検索語を AND で結合した検索式 (Query\_AND)
  - 検索要求文中のすべての検索語を OR で結合した検索式 (Query\_OR)
 に適合する文書についても、共起を含む文書と同様に 3.2 節の手法によって順位を補正する。
- (6) 検索結果が元の検索文に対する正解記事であれば正解として、再現率  $0, 10, 20, \dots, 100\%$  を満たすときの適合率を求める。ただし(2)で取り出したサンプル文書は評価対象から除く。また(2)で不要文書が得られない質問については評価の対象から除いた。

## 5. 実験結果と考察

手法 Combine, ID3, Add1, Add2 により共起を推定し、3.2 節の手法で Rocchio フィードバックと融合した結果を表 2 に示す。表中の数字は再現率  $0, 10, 20, \dots, 100\%$  の 11 ポイントでの適合率を平均したものであり、 $n$  は前章の実験手順(2)でのサンプル文書数である。

従来手法については、“Query”は検索要求文から作成したベクトルを用いた場合の検索精度(前章の実験手順(1)に相当)、“Rocchio”は Rocchio フィードバックによる検索精度を示す(実験手順(3)に相当)。“mod\_Rocchio”は改良 Rocchio フィードバッ

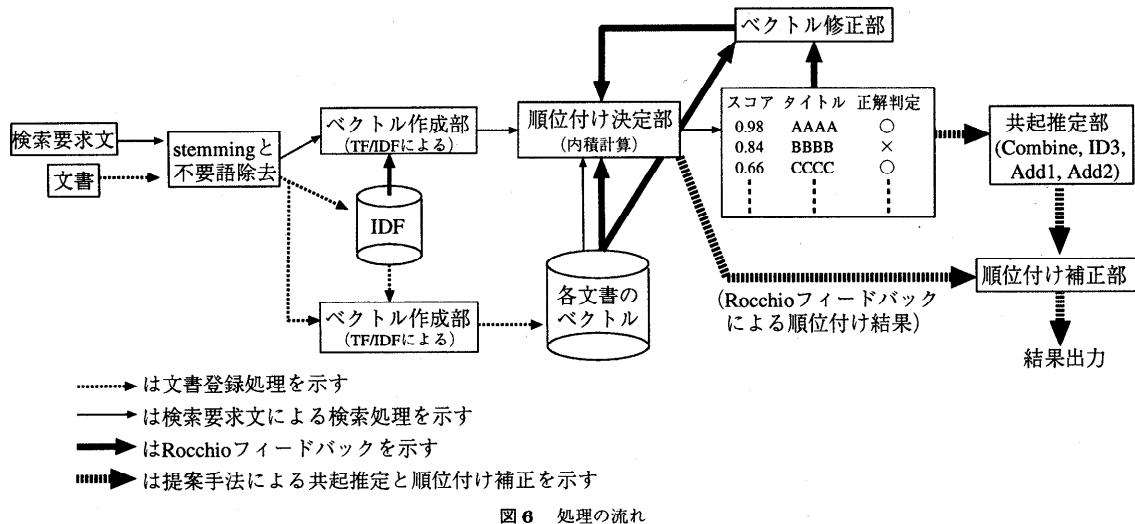


図6 処理の流れ  
Fig. 6 Processing flow.

表2 適合率平均 (%) (nはサンプル文書数)

Table 2 Average precision (%; n is the number of sample documents).

手法	n = 10	n = 30	n = 50
<b>Query</b>	13.4	9.2	8.0
<b>Rocchio</b>	17.9	15.9	16.1
<b>mod_Rocchio</b>	18.4	15.2	15.7
<b>Query_AND</b>	18.7	16.5	16.4
<b>Query_OR</b>	18.2	16.0	16.2
<b>Combine</b>	20.1	16.8	16.8
<b>ID3</b>	18.3	19.3	17.3
<b>Add1</b>	22.6	20.9	17.7
<b>Add2</b>	23.1	20.7	17.4

クによる検索精度を示す。

表2から**Query**に比べ**Rocchio**が優れていることが確認できる。**Rocchio**と**mod\_Rocchio**では,  $n=10$ では**mod\_Rocchio**の方が優れているが,  $n$ が大きくなると逆転している。

**Query\_OR**は検索要求文中のいずれか1つでも検索語を含む文書の順位を上昇させた場合の結果である。**Rocchio**と大差ない結果となっており, 検索要求文中の検索語を含む文書であっても必ずしも必要な文書とはならないことを示している。

共起を推定する各手法は  $n=10$ での**ID3**を除いて**Rocchio**, **mod\_Rocchio**に比べ優れた精度を示している。また**Query\_AND**と比較しても向上している。**Query\_AND**は検索要求文中のすべての検索語を含む文書の順位を上昇させた場合の結果である。これに対して共起を推定する各手法では, 重要な検索語の組合せを推定している。**Query\_AND**に比べて精度が向上しているため, 共起の推定によって他の検

索語と共にすることが重要ではない検索語を除くことができたことが分かる。

**Combine**はいずれのサンプル文書数でも**Rocchio**より優れた結果を示しているが, **Add1**と**Add2**には劣る。**Combine**で推定される共起は3.1節で述べた問題点(2)を持ち, 推定された共起は非サンプル文書中の必要文書に適合しないことが多い, 検索精度の向上効果が小さい。しかし問題点(1)を持つことは少ないため, 不要文書に適合する共起であることが小さく, 検索精度を悪化させる可能性は小さい。このため, サンプル文書数にかかわらず精度向上の効果があるものの, より正確に共起を推定できる方法に比べると精度向上効果は小さい。

共起を推定する各手法の中で, **ID3**はサンプル文書数  $n$  が小さい場合に低い精度を示している。一般に**ID3**で適切な決定木を作成するには十分な数の学習例を与える必要があることが知られているが, 共起の推定に用いた本例でもサンプル文書が少ない場合には不十分な結果しか得られないことが確認できる。検索者がレバансフィードバックを用いる場合, 多くの文書について必要ないし不要の判断をすることは稀であり, サンプル文書数  $n$  が小さい場合の対策は重要である。非サンプル文書を不要文書として用いることで学習文書数を増加させている**Add1**と**Add2**は, いずれのサンプル文書数でも優れた精度を示しており, 疑似的に学習文書数を増加させる提案手法がサンプル文書数が少ないのでの対策の1つになりうることを示している。

**Add1**, **Add2**は**Rocchio**や他の共起を推定す

表3 Rocchio フィードバックとの精度比較

Table 3 Improvement compared to the Rocchio feedback.

手法	比較	$n = 10$	$n = 30$	$n = 50$
<b>Query_OR</b>	向上	0.3/56	0.2/54	0.2/50
	悪化	-0.1/5	-0.2/7	-0.2/10
<b>Query_AND</b>	向上	15.5/4	10.7/5	9.6/3
	悪化	-7.1/1	-7.6/1	0.0/0
<b>Combine</b>	向上	12.7/15	9.7/8	9.1/7
	悪化	-2.3/10	-1.0/7	-0.5/4
<b>ID3</b>	向上	7.3/37	10.9/41	8.5/40
	悪化	-5.8/35	-4.8/34	-7.6/31
<b>Add1</b>	向上	9.9/44	11.4/44	8.2/42
	悪化	-3.8/23	-2.3/34	-5.3/39
<b>Add2</b>	向上	10.2/43	11.2/44	8.2/40
	悪化	-2.9/16	-2.9/30	-5.5/38

る手法より優れた結果を示しており、特に  $n = 10$ ,  $n = 30$  では **Rocchio** に比べて精度が 5% 程度向上している。決定木の作成をサンプル中の必要文書と不要文書が区別された段階で中止する **Add1** に対し、検索要求文中の全検索語を使って決定木を作成する **Add2** は検索式に含まれる検索語が多い。このため **Add2** は **Add1** に比べ少ない文書にヒットし、またヒットする文書は多くの検索語を含む。多くの検索語を含む文書は **Rocchio** フィードバックによって高いスコアが与えられるため、サンプル文書数  $n$  が大きくなると、**Add2** によって推定される共起を含む文書の多くがサンプル中に登場する。サンプル文書は精度評価の対象としないため、サンプル文書数  $n$  が大きくなるにつれ **Add1** に比べて **Add2** の効果は小さくなる傾向があるが、本実験では大きな差異は認められない。

表3 に各質問ごとの適合率平均を **Rocchio** フィードバックと提案手法との間で比較した結果を示す。表の比較の欄は提案手法による検索精度（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“手法 = **ID3**, 比較 = 向上,  $n = 10$ ”の欄は、**ID3** 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 37 件あり、適合率平均の上昇の度合は 37 質問の平均で 7.3% であったことを示す。

いずれの手法でも精度が悪化している質問があり、すべての検索要求について効果が得られるわけではない。しかし特に **Add1** と **Add2** は悪化する度合に比べて向上する度合が大きく、また向上する質問数が悪化する質問数に比べ多いため、平均的には **Rocchio** フィードバックに比べ良好な結果を得ることができる。

## 6. おわりに

検索語間の重要な共起関係を推定し、推定した検索語の共起を利用して **Rocchio** フィードバックによる順位付けを補正する手法を提案した。共起関係を推定する複数の手法を示し、これら手法による検索精度の向上を実験により示した。共起関係を推定する手法の中では、決定木学習アルゴリズム **ID3** に対して疑似的に負例を増加させた学習例を与える、共起関係を推定する手法 (**Add1** および **Add2**) の効果が大きい。サンプル文書数が少數の場合、これらの手法を用いることで **Rocchio** フィードバックに比べて適合率平均を 5% 程度向上することができた。

この効果は通常の検索文入力による検索 (**Query**) に対して **Rocchio** フィードバック (**Rocchio**) がもたらす精度向上効果と同程度である。**Rocchio** フィードバックは検索精度向上に有効な手法として認められていることから、提案手法による精度向上の効果は比較的大きいといえる。

今回用いたテストコレクションの中には検索要求文中の検索語を含んでいても、文書中の主題となっていないために正解文書とならない場合が見られた。提案手法では決定木作成に用いる検索語を選択する際に各文書内の単語の登場回数、出現位置などの情報は用いていないが、これらの情報は文書検索、および文書からのキーワード抽出に有効であり<sup>15), 16)</sup>、これらの情報を利用することでレレバансフィードバックの精度を向上できると予想する。

提案手法では推定した重要な共起を含むかどうかで文書の順位付けを修正したが、実際には共起ごとに重要度が異なると考えられる。共起の程度によって **Rocchio** フィードバックによるスコアを修正する、たとえば、多くの正解文書に含まれる共起や、重みの大きい検索語を含む共起については **Rocchio** フィードバックのスコアを大きく向上させる、等の方法により精度向上が図れると予想する。

検索語間の重要な共起を推定する手段として **ID3** を用いたが、**ID3** による決定木作成は学習例とその属性数（検索語数）の積に比例する計算量を必要とする。NPL テストコレクションを用いた実験では、学習例を大量に必要とする **Add1** および **Add2** では 1 質問あたり 1 分近い処理時間<sup>\*</sup>を必要とする場合があり、大量の文書を扱う文書データベースでは実用性の点で問題

\* Sun SparcStation 20 (hyperSPARC, 動作周波数 125 MHz, メモリ 224 MB) を実験に使用。

がある。学習例をランダムサンプリングすることで学習例を減らす方法<sup>10)</sup>や、単語の文書頻度（Document Frequency）を用いて検索式に適合するデータベース中の文書数を推定する手法を利用する<sup>6),17)</sup>など、近似手法により計算時間を短縮する効果と、検索精度への影響を検証する必要がある。

## 参考文献

- 1) Glasgow IDOMENEUS server,  
[http://www.dcs.gla.ac.uk/idom/ir\\_resources/](http://www.dcs.gla.ac.uk/idom/ir_resources/)
- 2) NPL テストコレクション,  
<ftp://ftp.cs.cornell.edu/pub/smart/npl/>.
- 3) Buckley, C., Salton, G. and Allan, J.: Automatic routing and ad-hoc retrieval using SMART:TREC2, *TREC-2*, pp.45–55 (1994).
- 4) Buckley, C., Salton, G. and Allan, J.: The effect of adding relevance information in a relevance feedback environment, *SIGIR*, pp.292–300 (1994).
- 5) Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill Advanced Computer Science Series, McGraw-Hill Publishing (1983).
- 6) Gravano, L., García-Molina, H. and Tomanic, A.: The effectiveness of Gloss for the text database discovery problem, Stan-CS-TN-93-2, Stanford Univ. (1994).
- 7) Harman, D.: Overview of the second text retrieval conference (TREC2), *The 2nd Text REtrieval Conference (TREC-2)*, pp.1–20, Department of Commerce, National Institute of Standards and Technology (1994).
- 8) Harman, D.: Overview of the 3rd text retrieval conference (TREC3), *The 3rd Text REtrieval Conference (TREC-3)*, pp.1–20, Department of Commerce, National Institute of Standards and Technology (1995).
- 9) Porter, M.F.: An algorithm for suffix stripping, *Journal of the Society for Information Science*, Vol.14, No.3, pp.130–137 (1980).
- 10) Quinlan, J.R.: *C4.5: Programs for machine learning*, Morgan Kaufman (1993).
- 11) Rissanen, J.: Modeling by shortest data description, *Automatica*, pp.465–471 (1978).
- 12) Rocchio, J.J.: Relevance feedback in information retrieval, *The SMART Retrieval System*, pp.313–323, Prentice-Hall (1971).
- 13) Spink, A.: Term relevance feedback and query expansion: relation to design, *SIGIR*, pp.81–90 (1994).
- 14) Pfeifer, U. and Huynh, T.: FreeWAIS-sf (1994).  
<ftp://ls6-www.infomatik.uni-dortmund.de/>  
pub/wais/freeWAIS-sf-1.0.tgz.
- 15) 原 正巳, 中島浩之, 木谷 強: 単語共起と語の部分一致を利用したキーワード抽出法の検討, 自然言語処理研究会資料, NL-106, 情報処理学会 (1995).
- 16) 高木 徹, 木谷 強: 単語出現共起関係を用いた文書重要度付与の検討, 情報学基礎研究会資料, FI-41-8, 情報処理学会 (1996).
- 17) 中島浩之, 木谷 強: 単語の文書頻度を利用した決定木学習アルゴリズムによる relevance feedback の高精度化, 情報学基礎研究会資料, FI-45-97, 情報処理学会 (May 1997).
- 18) 菅井 猛, 和田光教: WWW 上の電子新聞に対する情報フィルタリングとその評価, 情報学基礎研究会資料, FI-43-13, 情報処理学会 (1996).
- 19) 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).
- 20) 三輪真木子: データベースサーチャの視点, 情報処理, Vol.33, No.10, 情報処理学会 (1992).

(平成 10 年 3 月 27 日受付)

(平成 10 年 12 月 7 日採録)

### 中島 浩之（正会員）



1970 年生. 1994 年東京工業大学大学院理工学研究科情報工学専攻修士課程修了. 同年 NTT データ通信(株)(現(株)NTT データ)入社. 文書検索, キーワード抽出に関する研究開発に従事. 人工知能学会会員.

### 木谷 強（正会員）



1960 年生. 1983 年慶應大学工学部卒業. 同年日本電信電話公社(現日本電信電話(株))電気通信研究所入所. 1988 年 NTT データ通信(株)(現(株)NTT データ)に転籍. 形態素解析, 情報抽出, 文書検索に関する研究開発に従事. 工学博士.

### 岡田 守（正会員）



1972 年広島大学大学院修士課程修了. 同年日本電信電話公社(現日本電信電話(株))電気通信研究所入所. 1988 年 NTT データ通信(株)(現(株)NTT データ)に転籍. 1998 年より高知工科大学情報システム工学科教授. 工学博士. 電子情報通信学会会員.