# 協調フィルタリングにおける評価値予測アルゴリズムを応用した検索式拡張手法

帆足 啓一郎† 松本 一則†
井ノ上 直己† 橋本 和夫†

既存の検索式拡張手法では，適合文書集合に含まれる文書は，初期検索式との類似度にかかわらず，すべて同等に扱われる．このため，初期検索式との類似度が高い文書から抽出された単語と類似度が低い文書から抽出された単語が同等に扱われることになり，これらの単語を利用して拡張された検索式に基づく検索の精度の劣化の原因となっている．本研究では，協調フィルタリングで使用される評価値予測アルゴリズムを応用し，初期検索式に出現しない単語のスコアを「予測」することによって検索式拡張を行う手法を提案する．協調フィルタリングでは，評価対象ユーザとの相関が強いユーザの評価データを利用し，そのユーザのアイテム評価値を予測する．本手法を検索式拡張に応用することにより，拡張対象単語のスコア算出時に初期検索式との類似度が考慮されるため，拡張された検索式を利用した検索精度向上が期待される．本研究では，TREC データに対する評価実験を行い，提案手法の有効性を示す．

# Query Expansion Method Based on Predictive Algorithms for Collaborative Filtering

Keiichiro Hoashi,† Kazunori Matsumoto,† Naomi Inoue† and Kazuo Hashimoto†

In this research, we propose a novel query expansion method baseed on predictive algorithms used in collaborative filtering. Existing query expansion methods such as Rocchio's algorithm consider all documents in the relevant document set equally. This means that even in situations where the actual relevance of the documents in the set are not clear, documents with high similarity to the initial query are considered the same as documents with low similarity. In predictive algorithms, votes of users with high correlation to the active user are used to predict votes of the active user. By applying this algorithm to query expansion, jit is possible to emphasize information extracted from documents highly similar to the initial query, which is expected to lead to improvement of text retrieval. Through experiments conducted on TREC data, we have proved the effectiveness of the proposed method.

## 1. はじめに

高精度な情報検索システムにおいて不可欠な技術の1つとして「検索式拡張」（query expansion）があげられる．検索式拡張とは，情報検索システムのユーザが入力した入力文から生成される初期検索式に含まれる情報を自動的に拡張する手法である．検索式を拡張するための情報は，初期検索の結果得られた文書集合から抽出される．具体的には，ユーザ自身が初期検索で抽出された文書の適合性を判断し，システムに対してフィードバックを行う manual feedback[1]と，初期検索式との類似度が高い文書を適合文書と見なし，これを仮の適合文書集合（以下，仮適合文書集合とする）としてシステムにフィードバックする pseudo feedback[2]という2つの適合フィードバック（relevance feedback）手法が提案されている．検索式拡張は，これらの適合フィードバックの結果得られた適合文書集合あるいは仮適合文書集合に含まれる文書に含まれる単語の情報を抽出し，その単語を初期検索式に加えることにより実現される．

上記2つの適合フィードバック手法のうち，manual feedback は，初期検索の結果得られた文書に対し正確な適合性の評価が行われるため，検索式拡張がより有効になるという利点がある反面，適合性判断の負担がユーザにかけられてしまうという欠点がある．

† 株式会社 KDDI 研究所
KDDI R&D Laboratories, Inc.

一方，pseudo feedback ではユーザへの負担は軽減されるものの，フィードバックされる適合性の判断が完全ではないぶん，検索式拡張後の検索精度が manual feedback による検索式拡張に比べ劣化するという欠点がある．しかし，現実にはユーザからのフィードバック情報が得られるという状況が想定しにくいことなどから，現在では pseudo feedback による適合フィードバック手法に基づく検索式拡張が主流となっており，TREC[3)~7)]などの会議において数多くの研究成果が報告されている．

Rocchio のアルゴリズム[8)]に基づく検索式拡張手法など，既存の検索式拡張手法では，適合文書集合に含まれる文書はすべて同等に扱われている．すなわち，ユーザの要求に対する適合性の度合いについてはまったく考慮されておらず，ユーザの要求とぴったり合致する文書から得られる拡張対象単語と，ややユーザの要求からは外れているが，一応適合と見なされる文書から抽出される拡張対象単語が同等に扱われてしまう．また，pseudo feedback においては，仮適合文書集合に含まれる文書はあくまで類似度が高いために適合と見なされた文書の集合であるが，既存の検索式拡張手法では，ユーザの要求と合致している可能性の高い，すなわち初期検索式との類似度が高い文書と，ユーザの要求と合致していない可能性が高い，すなわち初期検索式との類似度が相対的に低い文書が同等に扱われるため，誤って仮適合文書集合に含まれた非適合文書から抽出された拡張対象単語が適合文書から抽出された拡張対象単語と同等の扱いで検索式拡張時に利用されてしまう．このことは拡張後の検索式による検索の精度の劣化の原因の 1 つとなっていると考えられる．

本論文では，上記の問題に対処するためには初期検索式との類似度を考慮した検索式拡張手法が必要であるという考えに基づき，協調フィルタリングにおいて利用される評価値予測アルゴリズムを応用した新たな検索式拡張手法を提案し，評価実験を通して提案手法の有効性を示す．まず，2 章において，協調フィルタリングおよびそこで利用される評価値予測アルゴリズムについて説明する．次に，3 章で既存の検索式拡張手法に関する説明を行い，その問題点を指摘する．次に，4 章で提案手法について詳しく説明する．5 章では，TREC データに基づく評価実験およびその結果を提示する．次に，6 章で関連研究との比較を行った後，7 章で本論文をまとめる．

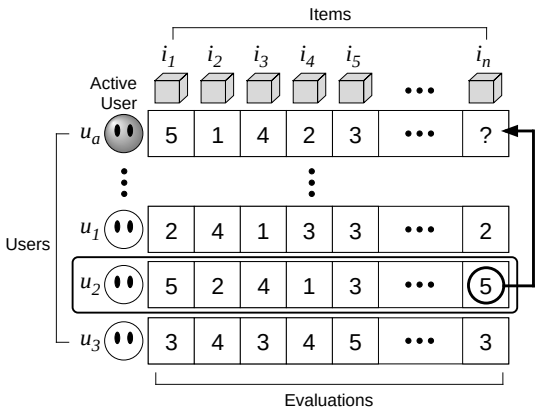


**図 1** 協調フィルタリング概念図
Fig. 1 Concept of collaborative filtering.

## 2. 協調フィルタリングにおける評価値予測アルゴリズム

本章では，協調フィルタリング（collaborative filtering），ならびに協調フィルタリングで使用される評価値予測アルゴリズムについて説明する．

### 2.1 協調フィルタリング

協調フィルタリングとは，あるアイテムに対するユーザの評価を，そのユーザの別アイテムに対する評価値と，他ユーザの評価データに基づいて予測するタスクである[9)]．具体的な適用例として，映画に対する多数のユーザの評価を入力とし，あるユーザがまだ見ていない映画に対する評価を，そのユーザの他の映画に対する評価と，他のユーザの評価に基づいて予測するタスクなどが考えられる．協調フィルタリングの根底にある考え方は，評価対象ユーザとの相関が高いユーザが高評価を与えたアイテムは，評価対象ユーザも高い評価をつける可能性が高い，というものである．つまり，ユーザの評価には相関があるということが協調フィルタリングの前提となっている．

**図 1** に協調フィルタリングの概念図を示す．図 1 では，評価対象ユーザ（active user）$u_a$ のアイテム $i_n$ に対する評価値を予測している様子が示されている．図 1 に示された $u_a$ 以外のユーザ $u_1$，$u_2$，$u_3$ のうち，$u_a$ と最も評価の傾向が類似しているユーザは $u_2$ であり，$u_2$ の $i_n$ に対する評価値が高いことから，$u_a$ の $i_n$ に対する評価も高いものと予測される．また，$u_1$，$u_3$ はいずれも $u_a$ とは評価の傾向が異なっており，$u_a$ との相関が低いことから，$i_n$ の評価値予測の際にはそれほど考慮されない．

協調フィルタリングにおける評価値予測アルゴリズムについては，これまで様々な手法が提案されている．

以下，代表的な評価値予測アルゴリズムについて説明する．

### 2.2 評価値予測アルゴリズム

Breese ら[10]によると，協調フィルタリングのための評価値予測アルゴリズムは *Memory-based* アルゴリズムと *Model-based* アルゴリズムの 2 つに大別される．Memory-based アルゴリズムは，すべてのユーザのすべてのアイテムに対する評価値を保持したうえで，評価値予測の際にはすべての評価値データを利用して予測値を算出するアルゴリズムである．一方，Model-based アルゴリズムは，評価値予測に先立ち，ユーザの評価データなどに基づき，ユーザ・アイテム・評価値を表すモデルを生成し，生成されたモデルと照らし合わせることによって評価値予測を行うアルゴリズムである．Memory-based アルゴリズムは手法自体がシンプルであり，容易に実装が可能であることや，新しい評価値データも容易に適用することができるが，評価値データの増加にともない，計算量が大幅に増加してしまうという欠点がある．Model-based アルゴリズムは，評価値データのモデル化ができれば少ない計算機資源で効率的に評価値予測を行うことができるが，モデル化自体に時間がかかってしまうほか，新たな評価値データをモデルに組み込む際にはモデルを再構築する必要があるという欠点がある．以下，それぞれのアルゴリズムについて説明する．

#### 2.2.1 Memory-based アルゴリズム

ここでは，Memory-based アルゴリズムについて説明する．まず，ユーザ $i$ のアイテム $j$ に対する評価値を $v_{i,j}$ とする．ユーザ $i$ が評価値を与えたアイテム群を $I_i$ とすると，ユーザ $i$ の平均評価値$\overline{v}_i$ は式 (1) によって定義される．

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \tag{1}$$

ただし，$|I_i|$ は $I_i$ に含まれるアイテムの数とする．Memory-based アルゴリズムでは，評価対象ユーザ $a$ のアイテム $j$ に対する評価値を，評価対象ユーザ $a$ の $j$ 以外のアイテムに対する評価値と，他のユーザの評価値に基づいて算出する．ここで，アイテム $j$ に対するユーザ $a$ の予測評価値を $p_{a,j}$ とすると，$p_{a,j}$ は式 (2) によって算出される．

$$p_{a,j} = \overline{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v}_i) \tag{2}$$

ただし，$n$ は $a$ 以外に評価を与えたユーザの数を示し，$w(a,i)$ はユーザ $a$ とユーザ $i$ の相関を表す値である．また，$\kappa$ は $w(a,i)$ の総和が 1 になるように設定された，正規化のための係数である．すなわち，$p_{a,j}$ とは，$a$ 以外のユーザの評価値を $w(a,i)$ によって重み付けした値の総和に基づいて算出される値である．$w(a,i)$ の算出方法については，いくつかの方法が提案されているが，式 (3) で表される相関係数と，式 (4) で表されるベクトル類似度による手法などが幅広く利用されている．

$$w(a,i) = \frac{\sum_i (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}} \tag{3}$$

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}} \tag{4}$$

このほか，Memory-based アルゴリズムの改良手法として，評価が与えられたアイテムが少ない場合，未評価のアイテムに対しデフォルト評価値を与えてから相関を算出する方法などが提案されている[10]．

#### 2.2.2 Model-based アルゴリズム

協調フィルタリングを確率的な観点からみると，評価対象ユーザに関する情報に基づき，アイテムに対する評価の期待値を算出するタスクであるともいえる．評価値の範囲が 0 から $m$ までの整数値であるとすると，$p_{a,j}$ は式 (5) によって算出することができる．

$$p_{a,j} = \sum_{i=0}^{m} Pr(v_{a,j} = i | v_{a,k}, k \in I_a) i \tag{5}$$

式 (5) の確率項 $Pr()$ は，ユーザ $a$ の $j$ 以外のアイテムに対する評価値が与えられた場合，評価値 $i$ が $j$ に与えられる確率を表す．この確率の算出方法としては，ベイジアンクラスタリングやベイジアンネットワークに基づく手法が提案されている[10]．

### 2.3 評価値予測アルゴリズム比較

Breese らは Memory-based アルゴリズムと Model-based アルゴリズムの比較のため，下記の評価データを利用した評価実験を行っている．

- **MS Web**
  Microsoft 社のウェブサイト内のユーザの訪問履歴を収集した評価データ．本データは，該当ページにユーザが訪問した場合は 1，訪問していない場合は 0 という評価値が記録されている．
- **Television**
  Neilsen 社から提供されたユーザのテレビ番組視聴データ．MS Web と同様，ユーザが見た番組については 1，見ていない番組には 0 という評価値が記録されている．
- **EachMovie**[11]

本データは，評価アイテムである映画に対し，ユーザが 0 から 5 までの評価値を与えたデータである．

以上のように，協調フィルタリングのための評価データは，評価値がバイナリで表される暗示的評価データ（implicit votes）と，ユーザが評価アイテムに対し数段階にわたる評価値を与える明示的評価データ（explicit votes）の 2 種類に大別することができる．上記の評価データのうち，MS Web と Television は暗示的評価データであり，EachMovie は明示的評価データである．

Breese らの報告によると，上記評価データのうち，MS Web と Television については Model-based アルゴリズムが Memory-based アルゴリズムを上回る結果が得られた一方，EachMovie に対する実験では逆に Memory-based アルゴリズムの方が Model-based アルゴリズムよりも優位であるという実験結果が得られている．この評価実験の結果から，明示的評価データについては Memory-based アルゴリズムの方が Model-based アルゴリズムより適しているという結論が得られる．

## 3. 従来手法

ここでは，既存の検索式拡張手法のうち，最も幅広く利用されている Rocchio のアルゴリズムについて説明し，同手法の問題点を明らかにする．

### 3.1 Rocchio のアルゴリズム

現在，最も有効な検索式拡張手法の 1 つに Rocchio のアルゴリズムに基づいた手法がある[8]．Rocchio のアルゴリズムは 1960 年代半ばに提案されており，現在に至るまで SMART[12]など数多くの検索システムに採用されている．

Rocchio のアルゴリズムは，検索対象文書などをベクトルとして表現するベクトル空間モデルに基づいている．元々は「ある入力文に対する適合性が既知の場合，その入力文を表す最適なベクトルとはその入力文に対する適合文書との類似度を最大にし，かつ非適合文書との類似度を最小化するものである」という思想に基づいて提案された手法であり[13]，この最適なベクトルは適合文書集合中の文書を表すベクトル群の重心と，非適合文書を表すベクトル群の重心との差分ベクトルであるとしている．Rocchio によると，ここでの最適なベクトル $\vec{Q}_{opt}$ は式 (6) によって表される．

$$\vec{Q}_{opt} = \frac{1}{R}\sum_{D\in Rel}\vec{D} - \frac{1}{N}\sum_{D\notin Rel}\vec{D} \qquad (6)$$

ただし，$R$，$N$ はそれぞれ検索対象文書集合中の適合文書，非適合文書の数を表し，$Rel$ は適合文書を表すベクトルの集合とする．式 (6) の計算の結果，値が負になったベクトルの要素はその値を 0 とする．

式 (6) の最適ベクトルの算出は，元の入力文のベクトルを適合文書を表すベクトルに近づけるとともに，非適合文書のベクトルから遠ざけるという効果を持つ．しかし，この過程では元の入力文を表すベクトルの特徴が反映されていない．そこで，最適ベクトルを算出する際に元の入力文のベクトルの特徴を取り入れた手法も開発されている[14]．修正された最適ベクトルの定義を式 (7) に示す．

$$\vec{Q}_{new} = \alpha\times\vec{Q}_{org} + \beta\times\frac{1}{R}\sum_{D\in Rel}\vec{D} - \gamma\times\frac{1}{N}\sum_{D\notin Rel}\vec{D} \qquad (7)$$

式 (7) には，元のベクトル，適合文書のベクトル，および非適合文書のベクトルの影響を調整するためのパラメータ $\alpha$，$\beta$，$\gamma$ が付与されている．SMART はこの式に基づく検索式拡張手法を使用しており，この手法により高い検索精度を実現している．

### 3.2 問題点

前記の説明から明らかなように，Rocchio のアルゴリズムでは適合/非適合文書集合全体に対しては $\alpha$，$\beta$，$\gamma$ といった係数によって重み付けが行われているが，適合文書集合に含まれている個々の文書については同等に扱われている．本来，ユーザからのフィードバック情報を検索式拡張時に効果的に活用するためには，ユーザの要求により適合した文書から得られた情報を強調する方が望ましいものと考えられる．しかし，Rocchio の手法では適合文書集合に含まれる文書はすべて同等に扱われてしまうため，このような考えは導入されていない．

また，適合フィードバックで pseudo feedback を使用した場合，仮適合文書集合に含まれている文書がすべて実際にユーザの要求と適合しているとは限らない．Pseudo feedback では，初期検索式との類似度が高い文書ほど実際にユーザの要求と適合している可能性が高いことが前提になっているが，Rocchio の手法では仮適合文書集合中の各文書と初期検索式の類似度の高低にかかわらず，仮適合文書集合に含まれる文書はすべて同等に扱われてしまうため，誤って仮適合文書集合に含まれた文書から得られた情報と，実際に適合している文書から得られた情報が同等に検索式拡張の際に利用されることになる．

以上の問題点から，拡張対象の情報が得られる個々の文書と初期検索式との類似度を考慮した検索式拡張

手法を導入することにより，さらに高精度な検索が実現できるものと期待される．

## 4. 提案手法

以下，本研究で提案する検索式拡張手法について説明する．

協調フィルタリングで使用される評価値データは，各ユーザの個々のアイテムに対する評価値を要素としたユーザベクトルの集合と見なすことができる．この場合，評価対象ユーザの未評価アイテムに対する評価値を予測するというタスクは，評価対象ユーザを表すベクトル内で値が含まれていない要素の値を予測することと同等である．

一方，情報検索では，各文書をベクトルで表現するベクトル空間モデルが広く利用されている[15]．ベクトル空間モデルでは，検索対象文書やユーザの入力文を表すベクトルの要素に各文書に出現する単語の重要度を表すスコアが格納されている．そして，ベクトル空間モデルにおける検索式拡張とは，初期検索式のベクトルに出現しない単語のスコアを算出するタスクであるとみなすことができる．

以上から，情報検索における初期検索式および適合文書集合をそれぞれ協調フィルタリングにおけるユーザと見なし，文書ベクトルに含まれる単語のスコアを評価値データの各アイテムの評価値と見なすことにより，協調フィルタリングで利用されている評価値予測アルゴリズムを検索式拡張に適用することが可能である．**図 2** に提案手法の概念図を示す．

図 2 では，図 1 における評価対象ユーザ $u_a$ を，初期検索式を表すベクトル $Q$ に置き換えている．検索式拡張の際は，初期検索式に出現していない単語，すなわちスコアが格納されていない単語（図 2 では単語 $t_n$）のスコアを算出するが，ここで，2.2 節で述べた評価値予測アルゴリズムを利用し，協調フィルタリングと同様，$t_n$ のスコアを「予測」することによって検索式拡張を行う．

本研究で提案する検索式拡張手法は，2.2 節で述べた 2 つの評価値予測アルゴリズムのうち，Memory-based アルゴリズムに基づくものとする．Memory-based アルゴリズムを選択した理由は，2.3 節で述べた Breese らによる評価実験の結果，明示的評価データに対する協調フィルタリングでは Memory-based アルゴリズムが Model-based アルゴリズムを上回る結果が得られたことである．協調フィルタリングにおける明示的評価データと暗示的評価データを比較した場合，暗示的評価データ内のアイテムにはバイナリの評価値しか

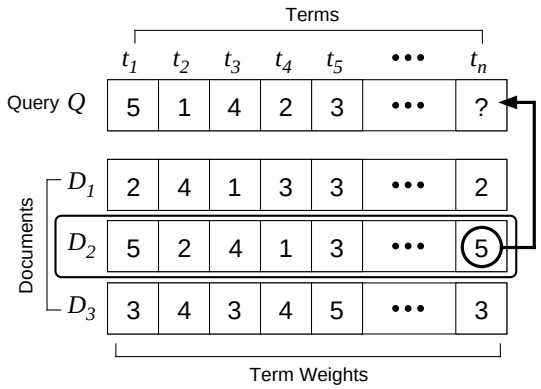


**図 2** 評価値予測アルゴリズムを利用した検索式拡張手法の概念図
Fig. 2 Concept of query expansion based on predictive algorithm.

与えられていないのに対し，明示的評価データには各アイテムに対しユーザが付与した数段階からなる評価値が与えられている．一方，ベクトル空間モデルにおいては，検索式ならびに検索対象文書を表すベクトルの要素には各単語のスコアが格納されている．単語のスコアがその文書内での単語の重要度を表す評価基準であると考えると，協調フィルタリングで利用されている 2 つの評価データのうち，明示的評価データの方がベクトル空間モデルに近いことは明らかである．したがって，協調フィルタリングにおける評価値予測アルゴリズムをベクトル空間モデルに基づく情報検索に適用する場合，Memory-based アルゴリズムの方が有効であると考えられる．

以下，提案手法について具体的に説明する．

初期検索式 $Q$ および検索対象文書 $D_i$ をそれぞれ $n$ 次元のベクトル $\vec{Q} = (q_1, \cdots, q_n)$ および $\vec{D}_j = (d_{j,1}, \cdots, d_{j,n})$ で表すとすると，拡張対象単語 $i$ のスコア $q_i$ を式 (8) によって算出する．

$$q_i = \overline{Q} + \kappa \sum_{k=1}^{K} Sim(Q, D_k)(d_{k,i} - \overline{D}_k) \quad (8)$$

ただし，$\overline{Q}$，$\overline{D}_j$ はそれぞれ $\vec{Q}$ および $\vec{D}_j$ の 0 以外の単語スコアの平均値を表す．また，$K$ は検索式拡張時に使用された検索対象文書数，$Sim(Q, D_j)$ は検索式 $Q$ と文書 $D_j$ の類似度をそれぞれ表すものとする．$Sim(Q, D_j)$ は式 (4) に示されたベクトル類似度によって算出する．式 (2) と式 (8) とを比較して明らかなように，提案手法は Memory-based アルゴリズムの式の $w(a, i)$ を，初期検索式と検索対象文書とのベクトル類似度に置き換えた手法である．

本手法の適用により，Rocchio の手法では考慮されていない初期検索式と検索対象文書との類似度が拡張

対象単語のスコア算出の式に含まれるため，初期検索式との類似度が高い文書から抽出された単語の情報が強調されることになる．このことにより，適合性の度合いに応じた拡張対象単語のスコア算出が可能となる．また，pseudo feedbackでは，仮適合文書集合に含まれる文書の実際の適合性が不明なまま検索式拡張が行われるが，本手法を利用することにより，初期検索式との類似度に応じて拡張単語のスコアが算出されるため，実際に初期検索式と適合している可能性が高いと考えられる類似度が高い文書からの情報が強調されることになる．このことにより，提案手法で拡張された検索式を利用した検索の精度向上が期待される．

## 5. 評価実験

以下，提案手法の評価実験について述べる．

### 5.1 目　　的

本実験の目的は，提案手法の有効性を示すことである．このため，本実験では，代表的な検索式拡張手法であるRocchioの手法との比較実験を行う．

### 5.2 評価データ

本実験では，TREC-6ならびにTREC-7のAd hoc Trackで使用されている評価データを使用する．いずれの評価データも，入力文として50個のTopicが用意されている．また，検索対象データはTRECのCD-ROM Vol 4，5に含まれている文書のうち，*Congressional Records*を除いた約53万件のテキスト文書から構成される．**表1**に本実験で使用するデータの詳細を示す．

また，上記データには，各Topicごとに適合する文書の一覧（以下，正解文書データ）が用意されており，評価結果の算出時に利用される．

### 5.3 使用システム

本実験で使用する検索システムでは，ベクトル空間モデルを利用する．入力文，ならびに検索対象文書に出現する単語の重みはTF*IDFによって算出される．ここで利用するTF*IDFの算出手法は，SMART[12]で利用されている手法に基づいたものである．以下，本実験で使用したTFおよびIDF値の算出式を示す．

- TF factor

$$\log(1 + tf_i) \tag{9}$$

- IDF factor

$$\log\left(\frac{M}{df_i}\right) \tag{10}$$

ただし，$tf_i$は単語$i$が文書内に出現した回数，$df_i$は検索対象文書のうち，単語$i$が出現する文書数，$M$は検索対象文書数をそれぞれ示すものとする．検索式（入力文）と検索対象文書との類似度は，前記のとおり，式(4)に示すコサイン値を算出することにより得ることとする．

**表1** TREC-6およびTREC-7 Ad hocデータ詳細
Table 1 Details of TREC-6 and TREC-7 ad hoc data.

| Data type | Data name | # of docs |
|---|---|---|
| TREC-6 Query | Topics 301-350 | 50 |
| TREC-7 Query | Topics 351-400 | 50 |
| Documents | *Financial Times*, *Federal Register*, *Foreign Broadcast Information Service*, *LA Times* | 528155 |

### 5.4 手　　法

本実験では，pseudo feedbackに基づく検索式拡張手法の評価を行う．まず，各Topicから生成された初期検索式を利用し，上位$K$件の文書を仮適合文書集合と見なし，検索式拡張を行う．また，提案手法ならびにRocchioの手法ともに，検索式拡張で初期検索式に加えられる単語数の上限を示すパラメータ$N$を設定し，算出されたスコアの上位$N$個の単語を拡張対象単語として，初期検索式に加える．なお，Rocchioの手法で使用されるパラメータ$\alpha$，$\beta$，$\gamma$は予備実験の結果，最適化された値である$\alpha = 1$，$\beta = 2$，$\gamma = 0$に設定した．

上記処理の結果，拡張された検索式を利用し，再度検索対象文書に対し検索を行った結果を最終結果とし，正解文書データとの照合により検索精度を算出する．

### 5.5 結　　果

以下，評価実験の結果を示す．

TREC-6データに対し，パラメータ$K$，$N$をそれぞれ$K = \{10, 20\}$，$N = \{10, 50, 100, 250, 500\}$と設定したうえで，提案手法（Col filter）ならびにRocchioの手法によって拡張された検索式に基づいた検索の平均精度（Average Precision）を**表2**に示す．なお，*Baseline*は初期検索式を利用して検索を行った結果である．また，同じ表に，Rocchioの手法を基準とした提案手法の改善率をあわせて記述する．

表2から明らかなように，提案手法による検索式拡張の結果，TREC-6の実験では*Baseline*と比較して最大22.6%の平均精度向上が得られている．また，**表3**に示されたTREC-7の実験結果からも，最大31.9%の平均精度向上が得られていることが分かる．

また，同じく表2に示された結果より，Rocchioの手法でも全般的に*Baseline*と比較して検索精度の向上が見られたものの，検索精度の向上率が最大14.0%にとどまっている．また，Rocchioの手法を基準とした提案手法の改善率を見ても，多くのパラメータ設定下

**表 2** TREC-6 データ実験の平均精度
Table 2 Average precision of experiments on TREC-6 data.

| | Col filter | | Rocchio | | Improvement rate | |
|---|---|---|---|---|---|---|
| $N$ | $K=10$ | $K=20$ | $K=10$ | $K=20$ | $K=10$ | $K=20$ |
| 10 | 0.2005 | 0.2046 | 0.1709 | 0.1724 | 17.3% | 18.7% |
| 50 | 0.2052 | 0.2217 | 0.1863 | 0.1915 | 10.1% | 19.3% |
| 100 | 0.2097 | 0.2285 | 0.1912 | 0.2004 | 9.7% | 8.3% |
| 250 | 0.2019 | 0.2171 | 0.2032 | 0.2075 | -0.6% | 6.8% |
| 500 | 0.1883 | 0.2061 | 0.2066 | 0.2125 | -8.9% | -3.0% |
| *Baseline* | 0.1864 | | | | | |

**表 3** TREC-7 データ実験の平均精度
Table 3 Average precision of experiments on TREC-7 data.

| | Col filter | | Rocchio | | Improvement rate | |
|---|---|---|---|---|---|---|
| $N$ | $K=10$ | $K=20$ | $K=10$ | $K=20$ | $K=10$ | $K=20$ |
| 10 | 0.1695 | 0.1764 | 0.1722 | 0.1693 | -1.6% | 5.4% |
| 50 | 0.1862 | 0.1933 | 0.1924 | 0.1869 | -3.2% | 3.4% |
| 100 | 0.1896 | 0.2006 | 0.1994 | 0.1943 | -5.9% | 3.2% |
| 250 | 0.1849 | 0.2048 | 0.2044 | 0.2015 | -9.5% | 1.6% |
| 500 | 0.1728 | 0.1964 | 0.2101 | 0.2069 | -16.5% | -6.5% |
| *Baseline* | 0.1553 | | | | | |

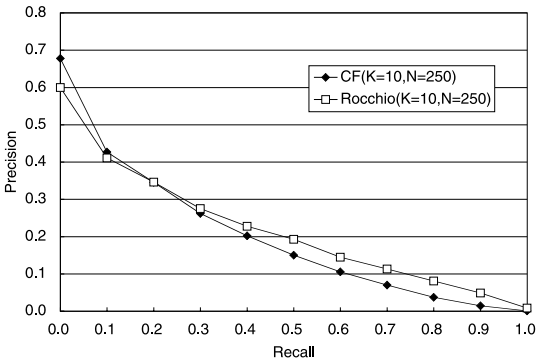


**図 3** TREC-7 実験での Recall-Precision 曲線（$K=10$, $N=250$）
Fig. 3 Recall-precision curveline of TREC-7 experiments ($K=10, N=250$).

で提案手法の検索精度が Rocchio の手法を上回っていることが分かる．この結果より，Rocchio の手法と比較しても提案手法が優位な結果を得ているといえる．

一方，表 3 に示された TREC-7 の実験では，$K=20$ と設定した場合は全般的に提案手法が Rocchio の手法を上回る平均精度が得られているものの，$K=10$ の場合は逆に Rocchio の手法の方が提案手法を上回る精度を得ていることが分かる．そこで，この結果をさらに詳細に分析するため，TREC-7 の各実験のうち，$K=10$，$N=250$ のパラメータ設定での提案手法ならびに Rocchio の手法の Recall-Precision 曲線を**図 3** に示す．

図 3 の Recall-Precision 曲線からも，全体的に Rocchio の手法の方が高い精度が得られていることが分かる．しかし，再現率（Recall）が低い状態では，逆に提案手法が Rocchio の手法を上回る適合率（Precision）を得ていることが明らかである．再現率が低い状態での精度とは，検索の結果，上位にランキングされた文書に対する精度を示す．実際に検索システムを利用するユーザのほとんどは，検索システムから出力された文書一覧のうち，上位の文書数件のみを閲覧するため，再現率が低い時点での精度が重要であることはいうまでもない．

そこで，各実験において，上位 5 件および 10 件での検索精度（Prec@5，Prec@10）を示す．**表 4**，**5** には TREC-6 データでの提案手法ならびに Rocchio の手法での Prec@5 および Prec@10，Rocchio の手法と比較しての提案手法の改善率を示す．さらに，「理想的な」ユーザによるフィードバックに基づく検索式拡張の精度を確認するため，manual feedback に基づき，Rocchio の手法で検索式拡張を行った際の Prec5, Prec10 も同様に示す．ただし，manual feedback は，上位 $K$ 件の適合文書をフィードバック情報として利用することとする．

また，**表 6**，**7** には，TREC-7 データでの同様の実験結果を示す．

表 4，5 から，TREC-6 の実験では 1 つのパラメータ設定を除き，いずれのパラメータ設定でも提案手法の精度が Rocchio を上回っていることが分かる．また，TREC-7 の実験の平均精度は提案手法と Rocchio の手法はほぼ同等であったが，表 6，7 で示された Prec@5 および Prec@10 は，TREC-6 同様，いずれのパラメー

**表 4** TREC-6 データでの Prec@5
Table 4 Precision at 5 documents for TREC-6 data.

| | Collaborative | | Rocchio | | Improvement rate | | Manual Rocchio | |
|---|---|---|---|---|---|---|---|---|
| $N$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ |
| 10 | 0.4360 | 0.4560 | 0.3680 | 0.3680 | 18.5% | 23.9% | 0.6480 | 0.6360 |
| 50 | 0.4080 | 0.4560 | 0.4040 | 0.4000 | 1.0% | 14.0% | 0.8080 | 0.7760 |
| 100 | 0.4560 | 0.4640 | 0.4000 | 0.4080 | 14.0% | 13.7% | 0.8560 | 0.8240 |
| 250 | 0.4280 | 0.4480 | 0.4120 | 0.4160 | 3.9% | 7.7% | 0.9160 | 0.8920 |
| 500 | 0.4040 | 0.4440 | 0.4160 | 0.4160 | -2.9% | 6.7% | 0.9320 | 0.9080 |

**表 5** TREC-6 データでの Prec@10
Table 5 Precision at 10 documents for TREC-6 data.

| | Collaborative | | Rocchio | | Improvement rate | | Manual Rocchio | |
|---|---|---|---|---|---|---|---|---|
| $N$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ |
| 10 | 0.3700 | 0.3880 | 0.3180 | 0.2980 | 16.3% | 30.2% | 0.5320 | 0.5380 |
| 50 | 0.3560 | 0.4020 | 0.3440 | 0.3380 | 3.5% | 18.9% | 0.6720 | 0.6680 |
| 100 | 0.3620 | 0.3820 | 0.3440 | 0.3540 | 5.2% | 7.9% | 0.7280 | 0.7260 |
| 250 | 0.3620 | 0.3920 | 0.3480 | 0.3580 | 4.0% | 9.5% | 0.7720 | 0.7680 |
| 500 | 0.3500 | 0.3940 | 0.3460 | 0.3580 | 1.2% | 10.1% | 0.7980 | 0.8040 |

**表 6** TREC-7 データでの Prec@5
Table 6 Precision at 5 documents for TREC-7 data.

| | Collaborative | | Rocchio | | Improvement rate | | Manual Rocchio | |
|---|---|---|---|---|---|---|---|---|
| $N$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ |
| 10 | 0.4000 | 0.3880 | 0.3480 | 0.3200 | 14.9% | 21.3% | 0.5720 | 0.5560 |
| 50 | 0.4360 | 0.4040 | 0.3340 | 0.3520 | 30.5% | 14.8% | 0.7800 | 0.7480 |
| 100 | 0.4360 | 0.4280 | 0.3920 | 0.3680 | 11.2% | 16.3% | 0.8440 | 0.8000 |
| 250 | 0.4360 | 0.4560 | 0.4120 | 0.3800 | 5.8% | 20.0% | 0.9080 | 0.8720 |
| 500 | 0.4320 | 0.4520 | 0.4120 | 0.4000 | 4.6% | 13.0% | 0.9200 | 0.8920 |

**表 7** TREC-7 データでの Prec@10
Table 7 Precision at 10 documents for TREC-7 data.

| | Collaborative | | Rocchio | | Improvement rate | | Manual Rocchio | |
|---|---|---|---|---|---|---|---|---|
| $N$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ | $K = 10$ | $K = 20$ |
| 10 | 0.3460 | 0.3540 | 0.3220 | 0.2980 | 16.1% | 18.8% | 0.5320 | 5040 |
| 50 | 0.3740 | 0.3780 | 0.3340 | 0.3340 | 12.0% | 13.2% | 0.6660 | 0.6520 |
| 100 | 0.3720 | 0.3800 | 0.3460 | 0.3400 | 7.5% | 11.8% | 0.7340 | 0.7080 |
| 250 | 0.3580 | 0.3740 | 0.3500 | 0.3480 | 2.3% | 7.5% | 0.7860 | 0.7620 |
| 500 | 0.3640 | 0.3800 | 0.3480 | 0.3520 | 4.6% | 8.0% | 0.7960 | 0.7900 |

タ設定でも Rocchio の手法を上回っていることが明らかになった．

提案手法の利点の 1 つとして，pseudo feedback で作られた仮適合文書集合に含まれる文書との類似度が拡張対象単語のスコア算出時に考慮されるという点をあげた．Rocchio の手法では，仮適合文書集合中の文書をすべて同等に扱うため，仮適合文書集合に誤って適合と見なされた文書が多く存在する場合，すなわち，初期検索の精度が低い場合は，拡張後の検索式を利用した検索精度が提案手法と比較して劣化すると考えられる．

この仮定を検証するため，TREC-7 の実験の初期検索で Prec@$K$（$K = 10$，20）が 0.5 以下の入力文の，提案手法ならびに Rocchio の手法で拡張された検索式を利用した検索の Prec@$K$ を比較する．**表 8** に $K = 10$ と設定したときに，提案手法と Rocchio の手法でそれぞれ拡張された検索式を利用した結果，Prec@$K$ が向上した入力文の数および割合を示す．また，**表 9** に，$K = 20$ の場合の同様のデータを示す．

表 8，9 の結果から，初期検索精度が低い入力文に対する検索式拡張の結果，提案手法が Rocchio の手法をすべて上回る検索精度を得ていることが分かる．仮適合文書集合中の実際の適合文書の存在比率が低い場合でも，Rocchio の手法では仮適合文書集合に含まれる文書を同等に扱ってしまうため，検索精度の向上は小さい．しかし，提案手法では初期検索式との類似度を考慮した検索式拡張を実現しているため，仮適合文書集合に含まれる適合文書が少ない場合でも Rocchio

**表 8** 検索式拡張後 Prec@10 が向上した入力文数（TREC-7）
Table 8 Number of topics which Prec@10 improved after QE (TREC-7).

| | Number of improved topics (Ratio) | |
|---|---|---|
| $N$ | Collaborative | Rocchio |
| 10 | 9 (23.7%) | 4 (10.5%) |
| 50 | 10 (26.3%) | 5 (13.2%) |
| 100 | 9 (23.7%) | 5 (13.2%) |
| 250 | 8 (21.1%) | 5 (13.2%) |
| 500 | 10 (26.3%) | 5 (13.2%) |

**表 9** 検索式拡張後 Prec@20 が向上した入力文数（TREC-7）
Table 9 Number of topics which Prec@20 improved after QE (TREC-7).

| | Number of improved topics (Ratio) | |
|---|---|---|
| $N$ | Collaborative | Rocchio |
| 10 | 14 (35.0%) | 5 (12.5%) |
| 50 | 13 (32.5%) | 9 (22.5%) |
| 100 | 15 (37.5%) | 9 (22.5%) |
| 250 | 12 (30.0%) | 8 (20.0%) |
| 500 | 11 (27.5%) | 8 (20.0%) |

の手法と比較して高精度な検索を行うことができることが，この分析の結果明らかになった．

以上の実験結果より，提案手法の有効性が示された．

### 5.6 考　　察

表 2 ならびに表 3 に示された実験結果より，Rocchio の手法では拡張対象単語数 $N$ が増加するにつれ，平均精度がほぼ単調に向上しているのに対し，提案手法では $N$ の増加に対する平均精度向上がみられず，その結果，$N$ が大きい条件下では提案手法の平均精度が Rocchio の手法のそれを下回る結果が得られている．ここでは，この実験結果について考察する．

検索式拡張を行うことによって得られる効果としては，適合文書をより多く検索ランクの上位に出現させる検索の効率化の効果と，より多くの適合文書の取得を目指す検索の網羅性向上の効果の 2 つがあげられる．検索の効率を向上させるためには，適切な単語情報を検索式に加えることが有効である．一方，網羅性の向上のためには，検索の効率化と同様，適切な単語情報の選択が重要であることはいうまでもないが，多くの適合文書を得るためには，そのぶん拡張対象単語数を増やす必要があると考えられる．

そこで，評価実験の結果を検証すると，提案手法は Rocchio の手法と比べて検索の効率向上効果を上げるのに秀でている一方，検索の網羅性を上げる効果については Rocchio の手法の方が提案手法を上回っていることが分かる．また，検索の網羅性を向上させるためには拡張対象単語数を増加させることが有効であるという仮定を立てたが，提案手法で拡張対象単語数を増加させても，Rocchio の手法ほど検索の網羅性が上がらないことも実験結果より明らかである．すなわち，提案手法では拡張対象単語のうち，算出されたスコアが下位ランクに位置付けられる単語については検索式拡張において有効なスコアが得られていないといえる．

そもそも，提案手法のもととなっている協調フィルタリングのタスクとは，未評価のアイテムに対するユーザの評価値を予測し，予測評価値の高いアイテムをユーザに推薦するというものである．すなわち，協調フィルタリングとは，多くのアイテムからなるデータベースの中からユーザにとって有用と思われる少数のアイテムを推薦するタスクであるといえる．したがって，検索式拡張に協調フィルタリングの評価値予測アルゴリズムを適用した場合，拡張対象単語数が少ないうちは協調フィルタリングと同様の効果が得られることは想定しやすく，実際に本研究での評価実験の結果からも提案手法が効果的であることが実証されている．しかし，Breese らの研究など，一般的な協調フィルタリングタスクでは，ユーザに対し 500 個ものアイテムを推薦するタスクは想定されていないため，本評価実験において拡張対象単語数が 250 や 500 といった非常に高い値に設定されている条件での評価値予測アルゴリズムの有効性は未知数である．このことから，拡張対象単語数が高く設定されている場合，スコアランク下位の単語に対するスコア算出は Rocchio の手法と比較して正確性を欠いている可能性があり，そのことがこの条件下での検索精度劣化につながっていると考えられる．本研究では検索の効率向上に重点をおいているため，検索の網羅性低下による平均精度の劣化については問題視していないが，スコアランク下位の単語に対するスコア算出方法については検討の余地がある．

## 6. 関 連 研 究

本研究と同様，pseudo feedback で得られる仮適合文書集合に含まれる文書の適合性の曖昧性を考慮した検索式拡張に関する研究報告として，Mitra らの研究報告があげられる[16]．Mitra らの研究では，適合フィードバック情報の精緻化を図るため，単語の共起情報などを利用して初期検索の結果得られた文書集合に含まれる文書の類似度を再計算し，仮適合文書集合内の文書の順位を変更してから Rocchio の手法により検索式拡張を行う手法である．本手法を適用することにより，従来手法を上回る検索精度が得られたと報告されている．

Mitra らが提案している検索式拡張手法と本研究の

提案手法を比較すると，pseudo feedback の問題について着目している点では類似している．しかし，Mitra らの手法では初期検索の結果得られた文書集合に含まれる文書について新たな類似度を算出する必要があるのに対し，本研究の提案手法では初期検索の際に算出した類似度をそのまま利用して検索式拡張を行っている．さらに，Mitra らの手法では，新しい類似度を算出するために単語の共起情報を取得する必要がある．以上の点から，Mitra らの手法と本研究の手法は検索式拡張によって同様の結果をもたらすものの，必要な計算量は本研究の手法の方が少なく，効率的な手法であるといえる．

また，Mitra らの研究報告では，pseudo feedback による悪影響の 1 つとして，検索式を拡張した結果，拡張された検索式の主旨が初期検索式の主旨とずれてしまう “query drift” をあげている．Mitra らは，特に初期検索精度が低い入力文では query drift が激しくなり，その結果，検索精度が劣化するという仮定を立てており，分析の結果，提案された手法を適用することにより低精度の入力文でも高い検索精度が得られることを実証し，低精度の入力文における query drift を抑えることができたとしている．本研究でも，表 8, 9 に示された結果により，初期検索精度が低い入力文においても提案手法が有効であることを示しているが，ここでは Mitra らと同様の query drift に関する分析を行うことにより，さらに提案手法の有効性を実証する．

まず，評価実験で使用した 100 個の入力文を，初期検索の上位 20 件以内に含まれる適合文書数に応じたグループに分類する．すなわち，初期検索の結果，上位 20 件の文書の中に適合文書が 1 個のみ存在するすべての入力文を 1 つのグループとし，同様に適合文書が 2 個，3 個…の入力文グループを作成する．次に，各グループごとに，初期検索，提案手法，ならびに Rocchio の手法のそれぞれにおける平均精度および Prec@20 の平均値を算出し，提案手法と Rocchio の手法の平均精度ならびに Prec@20 の初期検索からの改善率を算出する．

提案手法，ならびに Rocchio の手法における，各入力文グループの平均精度の改善率を**図 4** に示す．また，同様に提案手法と Rocchio の手法における Prec@20 の改善率を**図 5** に示す．これらのグラフの横軸は個々の入力文グループを表し，縦軸は初期検索と比較しての平均精度または Prec@20 の改善率を表す．改善率が負の値の場合は，検索式拡張の結果，平均精度または Prec@20 が初期検索を下回ったことを示す．なお，

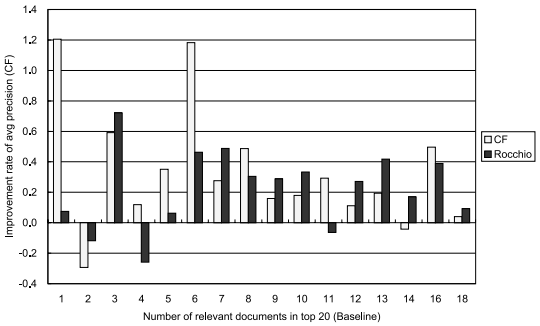


**図 4** 入力文グループごとの平均精度改善率
Fig. 4 Improvement rate of average precision per topic group.

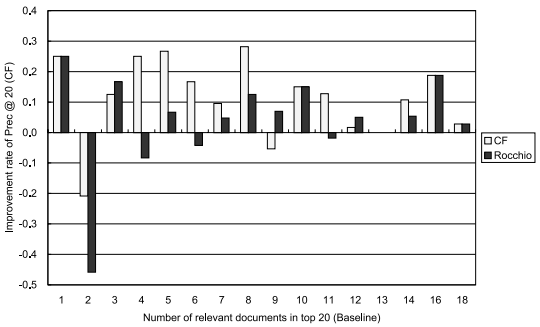


**図 5** 入力文グループごとの Prec@20 の改善率
Fig. 5 Improvement rate of Prec@20 per topic group.

該当する入力文が 1 つも存在しないグループはグラフより省略する．

まず，図 4 より，初期検索の精度が高い入力文については提案手法と Rocchio の手法における平均精度の改善率はほぼ等しいといえる．しかし，初期検索の精度が低い入力文については，提案手法における平均精度の改善率が Rocchio の手法の改善率をおおむね大きく上回っていることが明らかである．また，図 5 より，Prec@20 の改善率についても，同様の傾向があることが分かる．

以上の分析により，本研究の提案手法でも，Mitra の手法同様，初期検索精度が低い検索式の query drift を抑制する効果があることが明らかになり，低精度の入力文における提案手法の有効性が確認された．

## 7. ま と め

本論文では，協調フィルタリングで使用される評価値予測アルゴリズムを応用した新たな検索式拡張手法を提案した．

Rocchio の手法をはじめとする既存の検索式拡張手法は，検索式拡張時に使用される適合文書集合に含まれるすべての文書を同等に扱うため，pseudo feedback

を使用して仮適合文書集合を構築した場合など，適合文書集合中の文書の実際の適合性が不明な状況では非適合である可能性の高い文書も含めて適合文書として扱われてしまう．そのため，既存手法によって拡張された検索式を利用した結果，検索精度の劣化につながる可能性がある．

提案手法では，協調フィルタリング手法を検索式拡張に適用することにより，この問題の解決を図った．協調フィルタリングでは，評価対象ユーザと相関の高いユーザの評価情報を利用し，評価対象ユーザのアイテムに対する評価値の予測を行う．提案手法は，初期検索式との相関が高い，すなわち類似度が高い文書に含まれる単語の情報を利用し，初期検索式に含まれない単語のスコアを「予測」することにより，検索式拡張を行う手法である．

TREC データに基づく評価実験の結果，特に再現率の低い状態での精度では Rocchio の手法を上回る結果が得られた．また，初期検索の結果，検索精度が低かった入力文についても，提案手法によって検索式拡張を行った結果，従来手法を上回る検索精度が得られ，pseudo feedback における適合文書集合に含まれる文書の実際の適合性の曖昧性に効果的に対処できていることが明らかになった．以上の結果より，提案手法の有効性が示された．

**謝辞** 日頃ご指導いただく KDDI 研究所浅見所長に感謝いたします．また，本論文の評価実験において多大なご協力をいただいたスウェーデン・Uppsala 大学の Gustaf Brandberg 氏に感謝いたします．

## 参 考 文 献

1) Salton, G.: *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley (1988).
2) Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, M.: Okapi at TREC-3, Overview of the Third Text REtrieval Conference, pp.109–125 (1994).
3) Harman, D.: Overview of the Third Text REtrieval Conference, NIST SP 500–226 (1994).
4) Harman, D.: The Fourth Text REtrieval Conference, NIST SP 500–236 (1995).
5) Voorhees, E. and Harman, D.: The Fifth Text REtrieval Conference, NIST SP 500–238 (1996).
6) Voorhees, E. and Harman, D.: The Sixth Text REtrieval Conference, NIST SP 500–240 (1997).
7) Voorhees, E. and Harman, D.：The 7th Text REtrieval Conference, NIST SP 500–240 (1997).
8) Rocchio, J.: Relevance Feedback in Information Retrieval, *The SMART Retrieval System Experiments in Automatic Document Processing*, pp.313–323, Prentice Hall Inc.(1971).
9) Balabanovic, M. and Shoham, Y.: Fab: Content-based, collaborative recommendation, *Comm. ACM*, Vol.40, No.3, pp.66–72 (1997).
10) Breese, Heckerman and Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proc. 14th Conference of Uncertainty in Articial Intelligence*, pp.43–52 (1998).
11) http://www.research.digital.com/SRC/EachMovie
12) Singhal, A., Choi, J., Hindle, D., Lewis, D. and Pereira, F.: AT&T at TREC-7, *The 7th Text REtrieval Conference*, pp.239–251 (1999).
13) Singhal, A., Mitra, M. and Buckley, C.: Learning Routing Queries in a Query Zone, *Proc. SIGIR'97*, pp.25–32 (1997).
14) Salton, G. and Buckley, C.: Improving Retrieval Performance by Relevance Feedback, *Journal of the American Society for Information Science*, Vol.41, No.4, pp.288–297 (1990).
15) Witten, I., Moffat, A. and Bell, T.：*Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold (1994).
16) Mitra, M., Singhal, A. and Buckley, C.: Improving Automatic Query Expansion, *Proc. ACM SIGIR'98*, pp.206–214 (1998).

(平成 13 年 5 月 18 日受付)
(平成 14 年 1 月 16 日採録)

**帆足啓一郎**（正会員）

1995 年早稲田大学理工学部情報工学科卒業．1997 年同大学大学院修士課程修了．同年国際電信電話（株）入社．現在，(株) KDDI 研究所インターネットアプリケーショングループにて情報検索，情報フィルタリング等の研究に従事．

**松本 一則**（正会員）

1984 年京都大学工学部情報工学科卒業．1986 年同大学大学院修士課程修了．同年国際電信電話（株）入社，研究所所属．現在，KDDI 研究所インターネットアプリケーショングループにて，時系列データ処理，類似検索の研究開発に従事．特に実例からの知識獲得手法に興味を持つ．電子情報通信学会会員．

**井ノ上直己**（正会員）

1982 年京都大学工学部電子工学科卒業．1984 年同大学大学院修士課程修了．同年国際電信電話（株）入社．1987 年〜1991 年 ATR 自動翻訳電話研究所に出向．知識ベース，自然言語処理の研究に従事．1991 年より，KDDI 研究所において機械翻訳，音声認識，情報検索の研究に従事．工学博士．1991 年度学術奨励賞受賞．1995 年度日本音響学会技術開発賞受賞．電子情報通信学会，日本音響学会各会員．

**橋本 和夫**（正会員）

1977 年東北大学工学部電子工学科卒業．1979 年同大学大学院修士課程修了．同年国際電信電話（株）入社，研究所所属．現在，KDDI 米国研究所所長．自然言語処理，知識表現，エキスパートシステム等の研究開発に従事．情報科学博士．平成 12 年度電子情報通信学会論文賞受賞．電子情報通信学会，人工知能学会各会員．