グローバル分析とローカル分析に基づく検索支援

崔 超遠[†] 陳 漢雄[††]
古瀬 一隆[††] 大保 信夫[††]

初期検索式と関連があるキーワードをユーザに提供する問合せ修正手法は有効な情報検索技術である．既存の手法ではキーワードの共起関係あるいはシステムの順位付け技術によって提供する関連キーワードを決定するのが一般的であるが，前者では大量のキーワードを修正候補としてユーザに提供してしまうという問題があり，また後者ではユーザの検索意図を損なう恐れがある．この問題を解決するため，本稿ではグローバル分析とローカル分析を併用した検索支援手法を提案する．この手法ではまず，**カバー**という概念を用いて全キーワード集合を主要キーワードと非主要キーワードに分ける．検索支援の際には主要キーワード集合のみに対してローカル分析を行い，適切な数の修正候補に絞る．本稿では，絞り込み効果をもとにした主要キーワードの抽出手法および修正候補生成アルゴリズムを提案する．また，オンライン検索支援システムにこの提案手法を実装したプロトタイプを用いて，評価実験により提案手法の有効性と効率性を示す．

Query Refinement Based on Global and Local Analysis

Chaoyuan Cui,[†] Hanxiong Chen,[††] Kazutaka Furuse[††]
and Nobuo Ohbo[††]

Query Refinement is an effective information retrieval technique that interactively provides users with new keywords related to a particular query. Relevant keywords are generated by way of co-occurance or re-ranking mechanism of system. Lots of candidates will be occured according to the former because of co-occurance relationship, while candidates by re-ranking part of documents or keywords might lead to loss of user's intension. In order to solve these problems, we propose a query refinement method of synthesizing global and local analysis. The method use the concept *coverage* to divide the keyword database into two parts: the prime keyword and the non-prime keyword. During the refinement process, local analysis is performed to generate rational size of candidates with respect to prime keyword. We propose algorithms of prime keyword extraction and candidate generation based on screening effect, and implement an online refinement system to investigate the feasibility of our approaches. Moreover, experiments are conducted to confirm the effectiveness and efficiency of our proposed method.

1. はじめに

オンライン文献の急増とともに，ユーザはその中から目的の文献を選択することがますます困難になっている．実際，明確に与えられない問合せや，曖昧な言葉を含む問合せを行った場合，しばしば大量の非関連文献を検索してしまう結果となる．たとえば，*Google* は 42 億を超えるオンラインページを検索可能であるとしており，そこでのキーワード *database* による検索では 7 千万以上のページが結果として返される．ユーザには適合文献のみに出現するキーワードを前もって予測することができないため，最初から適切な問合せを作成することは非常に困難である．多くの検索システムにはキーワードの追加入力による絞り込みなどの問合せ修正の機能が提供されているが，ユーザ自身がキーワードを予測できない以上，効果的な修正は望めない．そこで，ユーザが問合せを作成する際に必要となる知識をあらかじめデータベースから自動抽出し，その知識を利用した検索支援システムが必要と考えられる．

インターネット上の検索エンジンも含め，現在の情報検索システムの多くはユーザの問合せに対して順

† 筑波大学システム情報工学研究科
Doctoral Program in Systems and Information Engineering, University of Tsukuba
†† 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics, University of Tsukuba

位付けした検索結果リストを返す．また，それらの一部は問合せ拡張やフィードバックなどの方法で修正候補となるキーワードを提供し，検索結果の絞り込みを行う．この分野の研究として，Mitra らによる自動的な問合せ拡張方法の提案がある[7]．このアプローチは検索した少量の文献を再ランキングすることによって初期検索を改善する．Carpineto らは上位文献に対して information-theoretic term-score 関数を使って修正候補にスコアを分配する方法を提案した[2]．この手法は，適合文献と全文献における確率分布を考慮し，その差を最大化するキーワードを選んで問合せ拡張を行う．

問合せ拡張は結果を改善するための有効な技術として認められている．しかし，問合せに不適切あるいは曖昧な言葉が含まれていると，膨大な文献が返されるという欠点がある．これに対してデータマイニングの技術を利用して結果を絞り込むアルゴリズムが提案された[4),11),12]．これは，段階的に結果を絞り込む方法である．このアプローチは情報の損失がないことを保証するが，そのためには巨大な共起キーワードのグラフ構造を構築しなければならない．また，Lexical Affinities（LAs）を利用して修正候補を提供するアルゴリズムも提案されている[5]．LAs とは，初期問合せ中のそれぞれのキーワードとそれに最も関連するキーワードの組である．絞り込みは，上位文献から LAs を抽出し，初期問合せに加えることによって行う．しかし，LAs はユーザの要望を正確に反映できるとは限らない．また，検索キーワードの選択に基づく支援方法[8),9)] は確かに有効であると考えられるが，いうまでもなく，前もってシソーラスを用意するか，検索時に動的にシソーラスが構築されなければならない．これは検索対象が巨大かつ動的であるような場合には空間的・時間的に効率的ではない．

問合せの修正を支援するため，Liu らはユーザに修正候補およびその絞り込み効果を提示する手法を提案した[11),12)]．Vélez らによる提案手法[1)] は修正候補を提示する点でこれと類似しているが，その手法では修正候補の構造を構築せず，修正支援も 1 回の提示のみで終わるシンプルなアプローチをとっている．近年，このようなアプローチに基づくインタフェースの Vivisimo [14)] など，一部の検索エンジンにおける実用的実装がみられる．大量な修正候補に対応するため，Chen らはさらに全キーワードを「代表」するための部分キーワード集合の抽出を試みた[4]．この方法ではすべてのキーワードを「代表」キーワードにマッピングし，共起関係を表すグラフ構造を構築する必要があり，上記に指摘したシソーラスに基づく方法の欠点を共有する．この欠点を克服するための手法は，Xu らによって提案された[6]．この提案において，その手法はグローバル分析とローカル分析に基づく方法であると主張されているが，グローバル分析に相当するのは既存の検索システムによって検索・順位付けを行う部分のみである．また，ローカル分析はグローバル分析の検索結果全体を用いる代わりにトップ n 件を用いて問合せ拡張を行うことにとどまっている．

以上の問題点を解決するために，筆者らは情報損失がないことを保証する問合せ修正支援手法を提案した[3]．本稿ではこの手法をさらに発展させ，グローバル分析に基づく「代表」キーワード抽出とローカル分析による修正候補作成を組み合わせた効率的な修正支援手法を提案する．また，提案手法を検証するため，以下の特徴を持つ検索支援システムを実装した例を示す．

- 問合せに関連する知識が少ないユーザに対して，システムは情報の損失がないことを保証しながら，不適切な初期問合せからの絞り込みを可能にする．
- ユーザに柔軟な絞り込み支援を提供する．すなわち，問合せの部分的な検索結果を提示するとともに，ユーザにとって自然な要求となる原則[4)] に従って適切な数の修正候補を提供する．

以下，2 章では本検索支援システムの概要を示し，筆者らが実装したプロトタイプシステムを紹介する．3 章においては，情報の損失がないことを保証するための重要な概念であるカバーと主要キーワードについて説明する．4 章では，グローバル分析に基づく主要キーワード抽出アルゴリズムを提案し，主要キーワードを用いて修正候補数を大幅に減少させることについて説明する．5 章では，ローカル分析に基づいた問合せ修正のための修正候補を生成するアルゴリズムを提案する．6 章では，テストデータに対する実験結果を分析する．これらの実験結果により，本手法の有効性，効率性を検証し，実際の実現可能性を確認する．

2. 構　　成

図 1 に検索支援システムの基本的な構成を示す．前述したグローバル分析とローカル分析に対応して，このシステムはオフラインフェーズとオンラインフェーズに分けられる．オフラインフェーズでは，キーワード抽出部が文献データベースからすべてのキーワードを抽出し，主要キーワード生成部がキーワードデータベースから主要キーワード集合を生成する．オンラインフェーズにおいては，まず，システムは文献データ

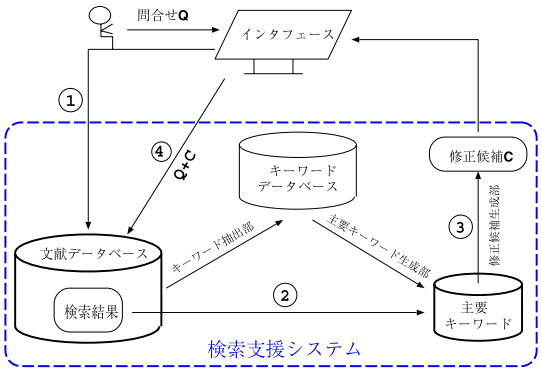

図 1 検索支援システムのアーキテクチャ
Fig. 1 The architecture of the refinement system.

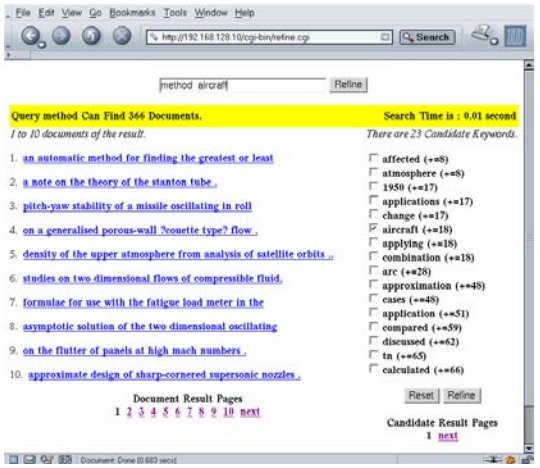

図 2 支援システムの例
Fig. 2 An example of refinement in prototype system.

ベースから問合せを含む文献を検索し，その検索結果に含まれる主要キーワードを抽出する．次に，修正候補生成部がユーザの要望を最大に満たす修正候補を選び，すべての候補キーワードおよび絞り込み効果を表す統計情報を同時に提供する．ユーザは統計情報に基づいて修正候補を選んで絞り込みを行う．この絞り込みのプロセスを，満足する検索結果が得られるまで繰り返し実行する．

図 2 はプロトタイプシステムの実行例である．ここでは文献数が 1,398 件，キーワード数が 4,612 個の CRAN データコレクションをデータベースとして使っている．この例では，ユーザが *method* というキーワードのみからなる初期問合せで検索を行っている．このシステムを用いて検索した結果，画面上には該当文献数 366 件と検索にかかったシステムの処理時間 0.01 秒が表示されるほか，文献へのリンク，修正候補数および候補キーワードが表示される．同時に，ユーザ自身による修正のためのヒントとして，各候補キーワードの絞り込み効果も表示される．たとえば，候補リストの最初の行 *"affected (+=8)"* は *affected* を初期問合せに追加すれば 8 件の文献に絞り込まれることを表している．ここでユーザが航空関係に興味があり，キーワード aircraft を選んだとする．すると新しい問合せは $\{method,\ aircraft\}$ になり，18 件の文献が結果として得られる．このように，修正された問合せはより明確になり，検索文献数が減少する．このシステムにより，ユーザは自分の検索したい文献が見つかればそれにアクセスすることができ，また結果に満足するまでこのプロセスを繰り返し実行することもできるという，柔軟な検索支援が可能になる．

3. 情報損失がないことを保証する概念

検索精度の向上に関してはさまざまな研究がなされているが，満足できる結果に至っていない．たとえば，地名，清涼飲料水の製品名，プログラミング言語の意味を持つ *Java* のような多義語に対してもユーザの検索意図を反映しようとするものとして，協調フィルタリングやパーソナライゼーションなどのアプローチがある．しかし，これらの方法に従って機械的に絞り込みを行った場合，ユーザの目的とするものが検索結果に含まれなくなる例が現れ，いずれ情報の損失が生じる．これに対して，筆者らは修正支援の段階ではすべての可能性をユーザに提示するいわゆる情報の損失がないアプローチをとる．本章では，情報の損失がないことを保証しつつ提示候補数を極力に抑えることを可能にした主要キーワードの概念を説明する．

まず，以下の議論に使われる記号について説明する．すべての文献からなる集合を $\mathbb{D}$ と表す．文献集合 $\mathbb{D}$ に含まれるすべての文献から抽出されるキーワードの集合を $\mathbb{K}$ と表す．ρ は文献 $d(\in \mathbb{D})$ のキーワードを抽出する関数であり，$\rho(d) = \{k|(k \in \mathbb{K}) \wedge k$ が d のキーワードである$\}$ と定義される．さらに，この関数を文献集合 $D \subset \mathbb{D}$ に対して拡張し，$\rho(D) = \bigcup_{d \in D} \rho(d)$ と定義する．明らかに $\rho(\mathbb{D}) = \mathbb{K}$ である．問合せとは，あるキーワード集合 $Q(\subset \mathbb{K})$ が与えられたときに，Q に含まれるすべてのキーワードを含む文献を全文献集合 $\mathbb{D}$ から求めることであり，次の関数 $\sigma(Q)$ のように定義される．$\sigma(Q) = \{d \mid d \in \mathbb{D} \wedge Q \subseteq \rho(d)\}$．以下の議論では便宜上，$|\sigma(Q)|$ を Q のサポートとよび，$Spt(Q)$ と書く．

Q の修正候補も $\mathbb{K}$ の部分集合である．一般に，Q および Y を $\mathbb{K}$ の部分集合とするとき，修正ルール $Q \Rightarrow Y$ は Y が Q の修正候補になることを表す．修正ルー

ルのサポート（Spt と略す）と信頼度（Cnf と略す）を次のように定義する．$Spt(X \Rightarrow Y) = |\sigma(X \cup Y)|$, $Cnf(X \Rightarrow Y) = |\sigma(X \cup Y)|/|\sigma(X)|$.

ここで，ルールのサポートとキーワード集合 (Q) のサポートは本質的に同じものであることに注意されたい．また，定義よりサポートは X に対して修正候補 Y を追加した後の検索文献数を表す．したがって，サポートが非常に大きいキーワード集合はほとんど stop word しか含まないことが分かる．一方，信頼度は $[0\ ,1]$ の範囲の値をとり，X に対して Y を追加する前後の検索文献数の比率を表す．直観的に，信頼度が 1 に近いほど Y の X に対する絞り込みの効果が薄いことが分かる．逆に，信頼度が 0 に近いものは絞り込み効果が顕著であるが，後に述べるように，信頼度の低さはカバーを保証するために必要となるキーワード（修正候補）の増加につながる．修正候補が多過ぎると，ユーザにとっては修正候補を選ぶ負担が大きくなる．一般に修正候補として好ましいものの条件として Chen らは次の諸基準を提唱した[4]．

(1) 絞り込み効果がある．
(2) 修正候補が多くない（結果的に修正も少ないステップ数で終わる）．
(3) システムが勝手に問合せの検索結果を狭めない．

絞り込み効果と修正候補数を考え，検索支援システムを構築するために，本稿では最小サポート[10] のほかに，最大サポート，最小信頼度と最大信頼度を導入し，以下の基本条件を満たすことを修正候補の必要条件とする．

$\theta_{s_l} < Spt(X \Rightarrow Y) < \theta_{s_u}$，および
$\theta_{c_l} < Cnf(X \Rightarrow Y) < \theta_{c_u}$

ここで，θ_{s_l}，θ_{s_u}，θ_{c_l}，θ_{c_u} はそれぞれ最小サポート，最大サポートおよび最小信頼度，最大信頼度を表す．

システムが勝手にユーザによって与えられた問合せの検索結果を狭めないことを保証するため，次の「カバー」および「主要キーワード」の概念を導入する．

定義 1：$\mathcal{K}(\subseteq 2^{\mathbb{K}})$ を $\mathbb{K}$ のべき集合の部分集合とし，$D(\subseteq \mathbb{D})$ を文献集合とする．次式

$$D \subseteq \bigcup_{Q \in \mathcal{K}} \sigma(Q)$$

が成り立つとき $\mathcal{K}$ は D の**カバー**である．同様に，

$$\bigcup_{Q \in \mathcal{K}'} \sigma(Q) \subseteq \bigcup_{Q \in \mathcal{K}} \sigma(Q)$$

が成り立つとき，$\mathcal{K}$ は $\mathcal{K}'$ $(\subseteq 2^{\mathbb{K}})$ のカバーである．

$\mathcal{K}$ が D のカバーであるとき，本稿では「$\mathcal{K}$ は D をカバーする」という表現も用いる．なお，この定義より，$2^{\mathbb{K}}$ は $\mathbb{D}$ のカバーであり，$2^{\rho(D)}$ は D のカバーとなる．

$\mathcal{K}$ が D のカバーであり，$\mathcal{K}$ の任意の真部分集合が D のカバーではないとき，$\mathcal{K}$ を D の**極小カバー**という．極小カバーを導入する目的は，キーワード集合から冗長なキーワードを取り除くことにある．一般に，特定の文献集合の極小カバーは一意ではない．本稿では，上に示した文献 4) の諸基準が修正支援を目的とする主要キーワードの生成に際してユーザの自然な要求をよく反映していると考え，それに従って極小カバーを求める．

定義 2：$\mathbb{K}$ を全キーワード集合とし，$\mathbb{D}$ を全文献集合とする．$\{\{k\} \mid k \in \mathbb{K}_p\}$ が $\mathbb{D}$（あるいは $2^{\mathbb{K}}$）の極小カバーであるとき，$\mathbb{K}_p$ を $\mathbb{D}$（あるいは $\mathbb{K}$）の**主要キーワード集合**という．また，主要キーワード集合に含まれるキーワードを**主要キーワード**という．

カバーの定義より，主要キーワード集合 $\mathbb{K}_p$ は以下の式を満たす．

$$\mathbb{D} = \bigcup_{k \in \mathbb{K}_p} \sigma(\{k\}) \equiv \bigcup_{K \subset \mathbb{K}_p} \sigma(K)$$

この式から，任意の文献が 1 つ以上の主要キーワードを必ず含んでおり，特定の主要キーワード集合から任意の文献が検索できることが分かる．主要キーワード集合は全文献をカバーするから，これに基づく検索支援システムでは情報の損失がないことが保証される．

4. グローバル分析：主要キーワードの生成

グローバル分析では，前章の概念に基づいて文献データベースから主要キーワード集合を生成し，主要キーワード間の関係から修正候補の構造を構築する．以上の過程は理論上，情報損失がないことを保証する概念のみで自動的に構築できるが，文献データベースの性質によっては実用性が損われる可能性がある．文献の著者などにより事前に与えられたキーワードが利用されるような文献データベースでは，サポートの非常に大きいキーワードもしくは非常に小さいキーワードしか含まないという，outliers（外れ文献）が存在することがあるが，このような文献は提案手法ではうまく対処できない．しかし，これらの outliers を具体的に見ると，前者はいわゆる stop word 同然の単語のみをキーワードにしているため，うまく処理できる方法はもともと存在していないと考えられる．一方，後者の場合は，含まれるキーワードのサポートが小さいため，最初から少数の文献しかヒットせず，それ以上の絞り込みなどの検索支援は明らかに不要と考えられる．本稿の実験では既存のデータベースを利用しているが，前処理で提案手法の本質を損なうことなく特定

のサポート範囲外の文献を outlier と見なして検索支援から排除する．当然のことながら，outlier は修正候補の提示による支援が行われないだけで，文献が直接ランク付けしてユーザに提供される方式でシステムからは検索可能である．なお，本稿の範囲を超えるが，文献からキーワードを抽出する段階でこの問題を意識して抽出を行うのであれば，この問題は最初から回避できるものと思われる．

主要キーワード集合を生成するアルゴリズムは，文献 4) においても 2 つ考案された．

1 つは，主要キーワード集合のサイズを適切に保ち，サポート値のばらつきをおさえるために，キーワードの中間サポート値を採用する発見的な方法である．この方法ではまず，中間サポート値を持つキーワードを主要キーワードとして選択し，全文献集合からこのキーワードを含む文献を削除する．次に中間サポート値（あるいはそれに最も近い値）を持つ別のキーワードを主要キーワードとして選択し，それを含む文献を全文献集合から削除する．文献集合が空になるまでこれを繰り返し，主要キーワード集合を決定する．

もう 1 つの方法は，中間サポート値の代わりに最小サポート値を用いる方法である．まず，すべての文献からそれぞれの文献の中で最小のサポート値を持つキーワードを 1 つずつ選ぶ．このようにして得られたキーワード集合は全文献集合のカバーになる．次に，サポート値によってこの集合のキーワードを昇順にソートする．最後に，ソート順にキーワードを調べ，それ以外のキーワードだけでも全文献をカバーできる場合には，そのキーワードを集合から削除して極小化する．最終的に残ったキーワードの集合が，主要キーワード集合となる．これらのアルゴリズムは，効率は良いものの，キーワード間の共起関係しか用いていないため，生成された主要キーワードは修正支援目的に最適とはいえない．

理論的に，ベクトルモデルでは索引キーワードが相互に独立であると仮定されている．しかし実際には，キーワード間の依存関係を無視して全文献集合に適用すると，支援効果が薄れてしまうことが指摘されている[8]．効果的な修正支援を行うためには，キーワード間の関係を利用すべきである．そこで，本稿では次の**絞り込み係数** RC（Refinement Coefficient）を導入する．

$$RC(k,d) = \frac{tf(k,d)}{|\rho(\{d\})|} \times \frac{\sum_{k_x \in \rho(\sigma(\{k\}))-\{k\}} cnf(k_x \Rightarrow k)}{|\rho(\{d\})|-1}$$

ここで $tf(k,d)$ はよく知られている係数であり，d に含まれる k の頻度を示す．$|\ \rho(\{d\})\ |$ は d に含まれるキーワード数である．また，3 章の定義により，$\rho(\sigma(\{k\}))$ は k と共起関係があるキーワード集合を表している．したがって RC は，共起関係にある他のキーワードに対する k の絞り込み効果の平均を表す．RC を用いて，次のような主要キーワード集合を生成するアルゴリズムを考案した．このアルゴリズムではまず，各文献から RC が最大になるキーワードを選ぶ．3 章の定義により，結果は明らかに全文献のカバーになる．次に，冗長なキーワードを除去し，得られた極小カバーを主要キーワード集合として出力する．以下にその疑似コードを示す．

アルゴリズム（主要キーワード生成）：
Input：文献集合 $\mathbb{D}$，キーワード集合 $\mathbb{K}$，サポート範囲 $MinSpt$ および $MaxSpt$
Output：主要キーワード集合 $\mathbb{K}_p$

```
1  K_p := ∅
2  forall d ∈ D
3  begin
5      X_d = {k | k ∈ ρ({d}) ∧ MinSpt ≤ spt(k)
               ≤ MaxSpt}
6      select k_0 such that
               RC(k_0, d) = max_{k∈X_d}{RC(k, d)}
7      K_p := K_p ⋃ {k_0}
8  end
9  sort K_p in ascending order of RC
10 forall k_i ∈ K_p
11     if ⋃_{k∈K_p−{k_i}} σ({k}) = D then
12         K_p := K_p − {k_i}
```

このアルゴリズムでは各文献に含まれるすべてのキーワードに対して処理を行わなければならないため，計算コストは高い．しかし，この処理は前もってオフラインで実行するので，問合せ支援処理の速度にはまったく影響を及ぼさない．むしろ，計算コストをかけてこの段階で精度の高い情報抽出を行うことが，次章で述べる効率的なローカル分析を可能にすると考えられる．

5. ローカル分析：修正候補生成アルゴリズム

前述したように，問合せ Q に対して，システムは $\sigma(Q)$ の一部と修正候補を同時にユーザに提示する．主要キーワードだけからなる問合せに対しては，この方式のシステムは最も有効かつ能率的に絞り込みを支援する．実際には多くの場合においてユーザの発行する

問合せには主要キーワード以外のキーワードが含まれることになるが，修正支援の段階でユーザの問合せに対して主要キーワードのみを候補として提示することは可能である．前述したように，このアプローチのメリットは，カバーの概念によって情報の損失がないことが保証されることと，少数の修正候補の提示によって効率的な問合せ修正が可能になることである．

グローバル分析では文献データベース全体の統計情報に基づいて主要キーワードの抽出を行ったが，検索支援の際には初期問合せの検索結果のみに対するローカル分析を行い，修正支援の候補を生成する．この段階では，前述の各概念をローカル化して用いることが可能である．このようにすることのメリットとしては，処理の一貫性や効率化があげられる．

主要キーワードのローカル化は次のようになる．ある文献 d に含まれる主要キーワードの集合を $\rho_p(d)$ で表す．このとき，$\rho_p(\{d\}) = \rho(d) \cap \mathbb{K}_p$ が成り立つ．$\sigma(Q)$ から抽出した主要キーワード集合に基づいて，システムは絞り込みキーワードを求める．絞り込みを行うときにどんなキーワードがユーザの希望を満たすかは重要なポイントとなる．以下ではこれを評価する尺度として RC をローカル化し，修正候補を求めるアルゴリズムを提案する．

RC をローカル化した RC_l は Q に対する絞り込み効果を表し，以下のように定義される．

$RC_l(k,d) = \frac{tf(k,d)}{|\rho(\{d\})|} \times cnf(Q \Rightarrow \{k\})$，$d \in \sigma(Q)$

次に，修正候補の生成アルゴリズムについて述べる．比較のため，異なるヒューリスティックに従って 2 つのアルゴリズム *RankCover* および *MaxRC* を考案した．前者では初期検索結果から上位 r 件の文献を選ぶ操作 $\sigma_r(Q)$ が行われるが，その順位付け関数としては，次の式により計算される文献 d と問合せ Q の類似度を用いる．

$$sim(d,Q) = \frac{\sum_{k_x \in Q} RC(k_x,d)}{\sqrt{\sum_{k \in \rho(d)} RC^2(k,d) \times |Q|}}$$

$sim(d,Q)$ を用いて，$\sigma_r(Q)$ は次の条件を満たす集合として定義される．

$\sigma_r(Q) \subset \sigma(Q) \wedge$
$|\sigma_r(Q)| = r \wedge$
$\forall d_1 \in \sigma_r(Q), \forall d_2 \in (\sigma(Q) - \sigma_r(Q)),$
$sim(d_1,Q) \geq sim(d_2,Q)$

以下にそれぞれのアルゴリズムの疑似コードを示す．

アルゴリズム *RankCover*
Input: $\mathbb{K}_p$, $\sigma(Q)$, and ranking threshold r.
Output: Refinement Candidates C_a
1 $C_a := \rho_p(\sigma_r(Q)) - Q$
2 **while** $\sigma(Q) - \bigcup_{k \in C_a} \sigma(\{k\}) \neq \emptyset$
3 **forall** $d \in \sigma(Q) - \bigcup_{k \in C_a} \sigma(\{k\})$
5 **begin**
6 select k_0 such that
$cnf(Q \Rightarrow \{k_0\}) =$
$max_{k \in \rho_p(\{d\}) - Q}\{cnf(Q \Rightarrow \{k\})\}$
7 $C_a := C_a \cup \{k_0\}$
8 **end**

アルゴリズム *MaxRC*
Input: $\mathbb{K}_p$, $\sigma(Q)$
Output: Refinement Candidates C_a
1 $C_a := \emptyset$
2 **forall** $d \in \sigma(Q)$
3 **begin**
4 select k_0 such that
$RC_l(k_0,\ d) = max_{k \in \rho_p(\{d\}) - Q}\{RC_l(k,\ d)\}$
5 $C_a := C_a \cup \{k_0\}$
6 **end**
7 sort C_a in ascending order of RC_l
8 **forall** $k \in C_a$
9 **if** $\sigma(Q) - \bigcup_{k_i \in C_a - \{k\}} \sigma(\{k_i\}) = \emptyset$ **then**
10 $C_a := C_a - \{k\}$

アルゴリズム *RankCover* は $\sigma(Q)$ の検索結果における上位 r 個の文献だけに出現する主要キーワードの重要性を強調する．これに対し，アルゴリズム *MaxRC* は，まず検索したすべての文献から 1 つずつ主要キーワードを選び，$\sigma(Q)$ のカバーを生成する．その後，そのカバーに含まれるキーワードのうち他のキーワードでカバーされる冗長なキーワードを削除し，カバーを極小化する．cnf を適用するかしないかという違いはあるが，2 つのアルゴリズムの目的はいずれも主要キーワード集合の部分集合となる修正候補 C_a を生成することにある．これら 2 つのアルゴリズムはカバーの概念に基づいているので，情報の損失はない．従来手法[4] で生成した修正候補との違いは，本稿の手法では C_a が $\rho_p(\sigma(Q))$ の部分集合になっており，そのサイズが大幅に縮小されることにある．しかし，$\sigma(Q)$ はオンラインで計算する必要があるため，応答時間に若干の影響を与える．この影響については，次章の実

験によって評価を行う．

6. 実験と評価

ここでは，提案する支援システムの有効性と効率性を実験によって検証する．実験システムは Debian Linux（カーネル 2.4）の上に C++で実装した．ハードウェアには，2.4 GHz のペンティアム III プロセッサ，1 GB のメモリ，70 GB のハードディスクの IBM PC-AT 互換機を用いた．

実験データには CRAN と JICST を用いた．CRAN には，1,398 件の文献とそこから抽出された 4,612 個のキーワードのほか，225 個の回答付きサンプル問合せが含まれており，提案手法などの検索精度を検証するために用いることができる．サンプル問合せの平均長（すなわちキーワード数）は 8.97 であり，平均適合文献数は 8.16 である．一方，JICST は，科学技術文献速報（電気工学編）の 1995–2000 年の 668,944 件の文献と 30,741 個のキーワードを含む．JICST は文献検索のテストコレクションとして提供されているものではないため，ここでは提案手法の速度を検証するためにのみ用いている．

4 章で述べた理由により，本実験では，主要キーワードを求める前に，文献集合に対して outlier（外れ文献）除去の前処理を行う．この処理ではまず，あらかじめサポート値の範囲を設定し，文献に含まれるキーワードのサポート値がこの範囲にあるかどうかを調べる．もしも上記サポート値の範囲に入るキーワードが存在しない文献があれば，その文献は outlier と見なす．キーワードのサポート値の分布を考慮し，JICST と CRAN のサポート値の範囲をそれぞれ [500，5000]，[10，200] と定めた．その結果，JICST では 3,696 件の文献が outlier となったが，CRAN には該当する文献は 1 件もなかった．したがって，CRAN のみに対して行っている以下の検索精度に関する評価は，この前処理の影響を受けていない．

6.1 修正候補数と実行時間

まずはじめに，修正候補数と実行時間について評価するための実験を行った．**図 3** に，outlier を取り除いた文献集合に対して主要キーワード集合を生成するアルゴリズムを適用した結果を示す．この図は，各アルゴリズムによって生成された主要キーワード数をサポート値ごとにカウントしたものである．たとえば，*MaxRC* より生成された主要キーワードのうち，サポートが 1,000 のものは約 100 個，5,000 のものは約 600 個となっている．これにより，生成した主要キーワード集合のサイズは全キーワード集合のサイズの 15 分の 1 程度に減少したことが確認できる．これは，提案手法で生成される主要キーワードが 3 章であげた修正候補の諸基準を満足することを示している．

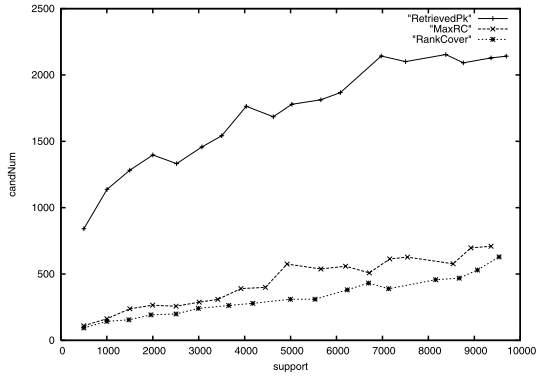

図 3 修正候補数
Fig. 3 Number of candidates.

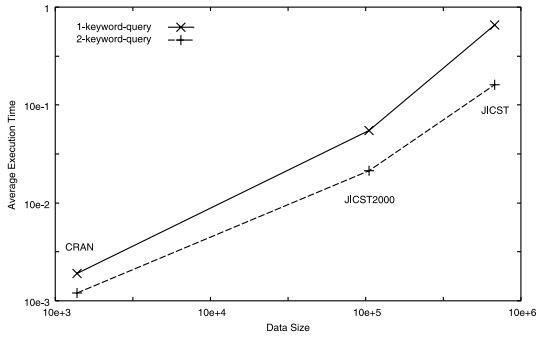

図 4 実行時間とデータベースサイズ
Fig. 4 Execution time vs. database size.

次の実験では，修正候補生成部が修正候補数と実行時間に与える影響について検証する．アルゴリズム *RankCover* では初期検索結果から上位 r 件を取り出して用いるが，この値には JICST に対しては 20 を，CRAN に対しては 3 を，それぞれ用いた．2 つのアルゴリズムの効率性と修正候補サイズの減少率を示すための比較対象として，$\sigma(Q)$ に含まれるすべての主要キーワードを選択する方法でも実験を行った．Carmel らの手法[5] は，主要キーワードの概念を用いてはいないが，そこで使われている LAs の概念ははこのような考え方に基づくと考えられる．以下では，この方法を *RetrievedPk* と表記する．

図 4 は，提案手法のスケーラビリティに関する実験の結果である．この実験では，中間サイズのデータの処理時間を考察するため，JICST のデータから 2000 年の 1 年分のみを取り出して独立したデータセットとして用いた．以降，このデータセットを JICST2000 と表記する．実行時間の比較は，CRAN，JICST，JICST2000

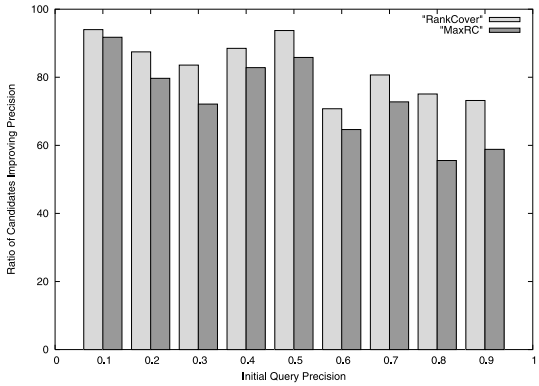

図 5 適合率を向上させる修正候補の割合
Fig. 5 Proportion of candidates improving precision.

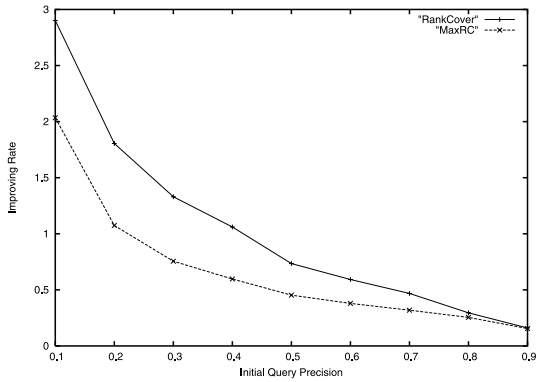

図 6 適合率向上率
Fig. 6 Improving ratio of precision.

の 3 つのデータセットに対して行った．図 4 より，実行時間とデータセットのサイズはほぼ線形的に比例する関係にあることが分かる．また，サイズが一番大きいデータセットでも実行時間は 1 秒未満であり，実験に用いたハードウェアの性能を考慮すればこの支援システムは十分実用可能であると考えられる．

6.2 主要キーワードによる適合率向上効果

修正候補生成部での処理により修正候補数は大幅に減少するが，これらの候補キーワードが適合率の向上に貢献し，絞り込み効果を発揮しなければ意味がない．これを確かめるため，次の実験では CRAN データセットを用いて絞り込みの正確さを調べる．CRAN が提供した問合せの平均長は約 9 であるが，提案手法を検証するためランダムにキーワードを組み合わせた問合せを用いた．なお，キーワード数が 3 を超える問合せの検索文献数はほとんど 0 になるため，それらの結果は以下に示していない．

再現率と適合率は，以下に示すよく知られている定義により計算される．

$$R = \frac{|Rel(Q) \cap \sigma(Q)|}{|Rel(Q)|},\ P = \frac{|Rel(Q) \cap \sigma(Q)|}{|\sigma(Q)|}$$

ここで，$Rel(Q)$ は問合せ Q の適合文献数を表す．

適合率の向上は不適切な初期問合せを修正してユーザの求める情報にどれだけ近づくことができるかどうかを測る最適な指標であると考え，これを個々の候補キーワードの有効性を評価する指標として用いた．実験ではまず，CRAN のサンプル問合せによって問合せを発行して初期適合率を計算する．次に，5 章で示したアルゴリズムによりすべての問合せの修正候補を求め，その修正候補を 1 つずつ初期問合せに追加して適合率を計算し，どの修正候補が初期適合率を向上させるかをテストする．

この実験の結果を**図 5** に示す．この図は，初期適合率を向上させる修正候補の全体に占める割合の平均を示したものである．この図から，初期問合せの適合率が 0 から 0.1 までの問合せに対して，アルゴリズム *RankCover* と *MaxRC* によって得られた修正候補の 9 割以上が初期問合せに対して適合率を向上させていることが確認できる．

適合率の増加の比率を調べた結果を**図 6** に示す．この図から，初期適合率が小さい問合せほど適合率の向上が顕著であることが分かる．特に，適合率が 0 から 0.1 までの場合に向上の度合いが高い．これは，本研究の目的である，適切な初期問合せを作成することが困難なユーザに対する検索支援効果が大きいことを意味する．

次の実験では，修正候補による適合率の向上効果について調べる．この実験では CRAN が提供するすべての問合せをあわせてキーワード集合と見なし，その中からサポートが 15 以上☆のものをランダムに選び，提案手法で生成する修正候補を追加して平均適合率の変化を確かめる．

図 7 は，初期問合せに対して 1 つまたは 2 つの修正候補を追加することによりどれほど平均適合率が向上したかを示したものである．この図から，修正候補となるキーワードを 1 つ追加すると平均適合率が顕著に向上することや，2 つのキーワードを追加する場合には平均適合率がさらに向上することが分かる．なお，テストコレクションのデータの性質により，本実験で 3 つ以上のキーワードを追加したほとんどの場合，検索結果が空になる．3 つ以上のキーワードを追加した場合の提案手法の有効性を検証するためには，より大きなテストコレクションを用いてさらなる実験を行う

☆ 15 という数字には直接的な根拠はないが，サポートが小さいキーワードについては修正候補の提供による支援は不要であるという outlier に関する筆者らの主張に基づいている．

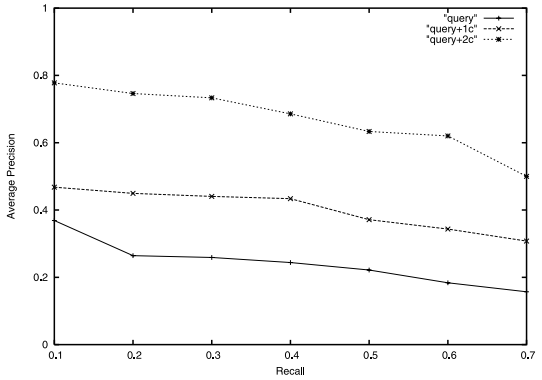

図 7 平均適合率と平均再現率
Fig. 7 Average precision and recall.

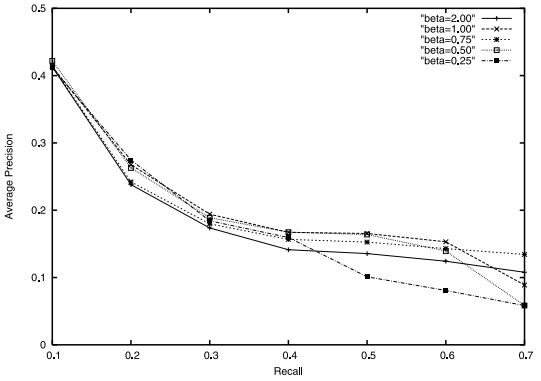

図 8 Rocchio におけるパラメータの影響
Fig. 8 Influence of parameters in Rocchio.

必要がある．

6.3 従来の修正方法との比較

ここでは，既存の問合せ修正手法と本稿の提案手法の比較を行う．多くのシステムは，以下に示す Rocchio の公式[8),13)] により問合せを拡張し，検索性能の向上を図っている．

Standard_Rocchio：

$$Q_{new} = \alpha Q_{orig} + \frac{\beta}{|D_r|} \sum_{\forall d_j \in D_r} d_j - \frac{\gamma}{|D_n|} \sum_{\forall d_j \in D_n} d_j$$

Ide_Regular：

$$Q_{new} = \alpha Q_{orig} + \beta \sum_{\forall d_j \in D_r} d_j - \gamma \sum_{\forall d_j \in D_n} d_j$$

ここで Q_{new} は拡張した問合せベクトルであり，Q_{orig} は初期問合せベクトルである．D_r と D_n はそれぞれ検索した文献における適合文献と不適合文献の集合を表す．適合文献に含まれる情報は不適合文献に含まれる情報より重要であることを考慮し，文献 8) に提案されているとおり，係数 γ を 0 に固定する．また，これも文献 8) に示されているとおり，上記の 2 つの方法における検索性能には大きな差がないことから，本実験では *Ide_Regular* のみを比較対象として採用する．

実験のステップは以下のようになる．

- $\sigma(Q_{orig})$ を求める．
- 類似度関数 $sim(d, Q_{orig})$ によって検索した文献をソートする．
- 適合文献を指定し，*Ide_Regular* で拡張式を求める．

まず，Rocchio におけるパラメータの影響を調べた結果を**図 8** に示す．ここでは α と β の相対関係を知ることが目的であるので，α は 1 に固定した．この図に示したように，β を 0.25 から 2 の範囲で変化させてみたところ，修正適合率に差異はほとんど見られないという結果となった．このため，本稿の提案手法と

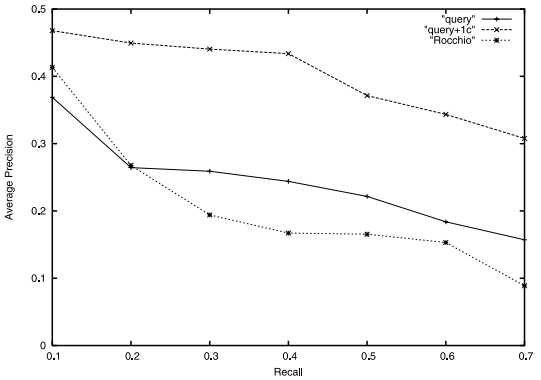

図 9 Rocchio との比較
Fig. 9 Comparison with Rocchio.

の比較には，最も良い結果が得られた $\beta = 1$ を用いることとした．

図 9 は Rocchio と本稿の提案手法を比較したものである．この図より，提案方法が顕著に適合率を向上させたことが分かる．一方，Rocchio は適合率向上の効果がほとんど見られないが，これは，初期問合せがそれぞれのキーワードを含むか含まないかの 2 値であり，これをベクトルとして表現したときにバイナリベクトルとなることの影響が大きいと考えられる．

7. おわりに

本稿ではグローバル分析技術とローカル分析技術を併用し，応答時間と修正候補の品質を向上させる検索支援技術を提案した．またプロトタイプ支援システムを実装し，複数のデータセットに対する実験により，提案アプローチの効率性，有効性および実用性を確認した．

今後は，提案手法を Web 検索エンジンとして機能させるための問題点を解決したいと考えている．本提案のアプローチではグローバル分析はオフライン処理

を前提に設計を行っているが，Web のような動的なデータ集合に対応するためには，オンライン検索に耐えうる程度の更新処理速度が要求される．また，本提案では統計情報をもとに主要キーワードの選択を行っているが，意味的に文献集合を代表し，かつ，ユーザの希望を満たすような主要キーワードの選択についても研究を進めたい．短期的には，実際のデータセットを用いて支援システムを実装し，提案手法をより詳細に検証する予定である．

参 考 文 献

1) Vélez, B., et al: Fast and Effective Query Refinement, *ACM SIGIR*, pp.6–15 (1997).
2) Carpineto, C., De Mori, R., Romano, G. and Bigi, B.: An Information-Theoretic Approach to Automatic Query Expansion, *ACM TOIS*, Vol.19, No.1, pp.1–27 (2001).
3) Cui, C., Chen, H., Furuse, K. and Ohbo, N.: Web Query Refinement without Information Loss, *APWeb2004*, LNCS 3007, pp.363–372 (2004).
4) Chen, H., Yu, J., Furuse, K. and Ohbo, N.: Support IR Query Refinement by Partial Keyword Set, *WISE*, pp.245–253 (2001).
5) Carmel, D., Farchi, E. and Petruschka, Y.: Automatic Query Refinement using Lexical Affinities with Maximal Information Gain, *ACM SIGIR*, pp.283–290 (2002).
6) Xu, J. and W.B. Croft: Query Expansion Using Local and Global Document Analysis, *ACM SIGIR*, pp.4–11 (1996).
7) Mitra, M., Singhal, A. and Buckley, C.: Improving Automatic Query Expansion, *ACM SIGIR*, pp.206–214 (1999).
8) Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, pp.24–31, Addison Wesley (1999).
9) Ali, S. and Crawford, R.: User-Thesaurus Interaction on a Web-based Database: An Evaluation of Users' term Selection Behaviour, *Proc. Infotech Oulu International Workshop on Information Retrieval*, pp.23–32 (2001).
10) Fayyad, U., Piatestsky, G. and Smyth, P.: From Data Mining to Knowledge Discovery in Databases, *The 3rd Knowledge Discovery and Data Mining*, pp.37–53 (1996).
11) Liu, Y., Chen, H., Yu, J.X. and Ohbo, N.: Using Stem Rules to Refine Document Retrieval Queries, *Proc. Inte'l Conf. on Flexible Query Answering System FQAS'98*, also in *LNAI, No.1495*, pp.249–260, Springer-Verlag (1998).
12) 陳 漢雄，劉 野，大保信夫：データマニングのキーワード検索に対する応用，*IPSJ SIG Notes*, Vol.97, No.64, pp.227–232 (1997).
13) Rocchio, G.C.: Relevance feedback in information retrieval, *The SMART Retrieval System — Experiments in Automatic Document Processing*, Salton, G. (Ed.), Prentice Hall Inc. (1971).
14) http://vivisimo.com/

(平成 16 年 6 月 20 日受付)
(平成 16 年 10 月 4 日採録)

(担当編集委員 吉岡 真治)

崔 超遠（学生会員）

1996 年中国内蒙古工業大学材料工程系卒業．2002 年筑波大学大学院理工学研究科修士課程修了．現在，同大学院システム情報工学研究科博士後期課程在学中．情報検索，データマイニングに関する研究に従事．

陳 漢雄（正会員）

1993 年筑波大学大学院工学研究科修了．同年同大学電子・情報工学系助手．1994 年つくば国際大学産業情報学科講師．2001 年筑波大学電子・情報工学系講師．博士（工学）．

古瀬 一隆（正会員）

1993 年筑波大学大学院工学研究科修了．（株）リコーソフトウェア研究所勤務，茨城大学工学部情報工学科助手を経て，1999 年筑波大学電子・情報工学系助手．博士（工学）．

大保 信夫（正会員）

1968 年東京大学大学院修士課程修了．同年同大学理学部助手．1980 年筑波大学電子・情報工学系講師．1995 年同大学電子・情報工学系教授．理学博士．