情報学基礎 61−27
自然言語処理 142−27
（2001. 3. 6）

関係学習を用いたフィルタ生成による対話的Webページ検索

岡部正幸　山田誠二
東京工業大学大学院総合理工学研究科知能システム科学専攻

概要：WWWの検索エンジンが返すヒットリストの上位には，ユーザが入力した単語との関連度が高いページがランクされるが，ユーザの興味と関係のないページが含まれていることも多い．本研究では，ユーザに訓練ページを示してもらうことにより，ユーザが必要とするページのみを提示するためのフィルタを関係学習に基づき生成する対話的検索システムを提案する．フィルタはキーワード，関係演算子，タグ情報で構成されるルール集合で表現され，ルールの条件部を満たすページのみがユーザに示される．このようなフィルタを生成するアルゴリズムを示すとともに，検索実験を行うことによってその効果を検証する．

Interactive Web Page Retrieval with Relational Learning based Filtering

Masayuki Okabe　Seiji Yamada
CISS, IGSSE, Tokyo Institute of Technology

Abstract: WWW Search Engines sometimes return undesirable hit lists which have many irrelavant pages because of the lack of information to specify the user's need. In this paper, we propose a system which removes irrelevant pages with query specific filter which is generated through the interaction with the user. The filter consists of several rules which are learned from training pages given by the user and made of keywords, relational operators and tags of html. We explain the algorithm for making the filter and show its performance on retrieval experiments.

1　はじめに

WWWの急速な普及によりインターネット上では日々多様な情報発信が行われている．検索エンジンは，これらWWW上に散在する膨大な量の情報へのアクセスを可能としており，WWWを情報源として活用する上で欠かせないツールとなっている．

検索エンジンは通常，ユーザから与えられる検索条件を用いて対象ページを絞り込み，それらをランキングしたものをヒットリストとして返す．しかし，ユーザが検索エンジンに入力する単語は一般的に平均2〜3語と少なく[2]，多くのWebページがヒットしてしまうため，それらを全て調べることは難しい．また，検索エンジンが提供する順位がユーザの検索意図を反映しておらず，順位が低くてもユーザの要求を満たすWebページ（適合ページ）がたくさん見つかる場合も多い．ランキング手法はそれぞれの検索エンジン独自の手法が用いられ，ほとんどの場合，その設定をユーザが調節することはできない．よって良い検索結果を得るには，検索エンジンの返すヒットリストから適合ページのみを自動的に選別する処理を行うことが必要である．

しかし，一般にユーザは検索を始める際に，目的とする情報を得られるWebページがどのような特徴を持っているかを明確には知らない．また，あるWebページが目的に沿うものであるかどうか，つ

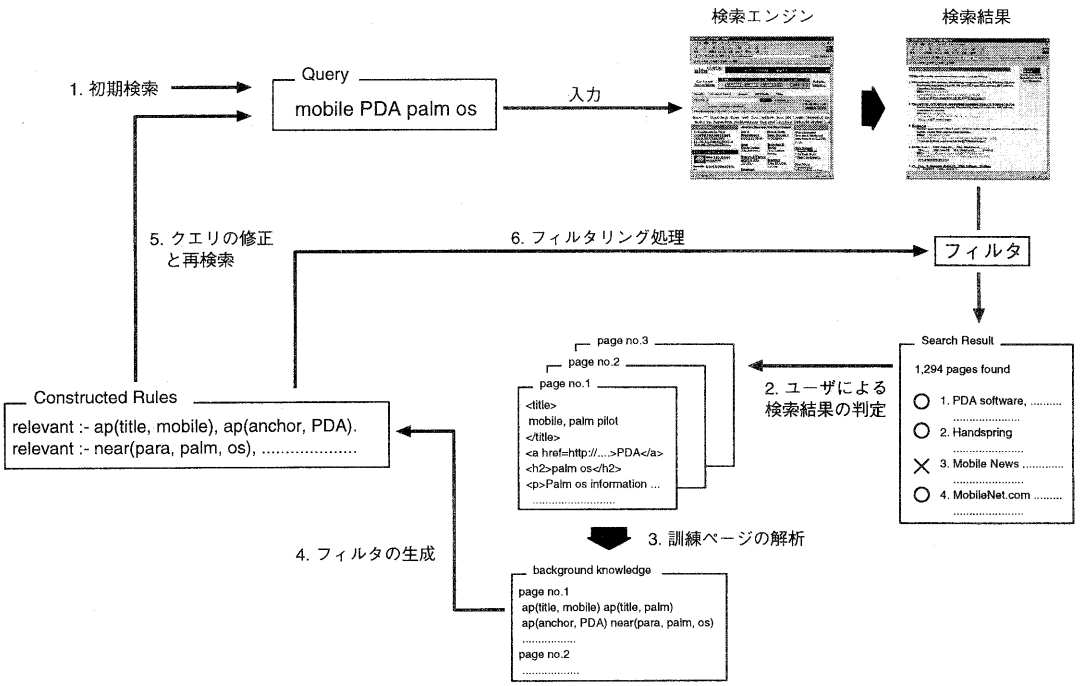

図 1　フィルタリングを伴う検索処理

まり適合ページであるか否かを判断することができても，その理由を判別条件として提示することは負担のかかる作業であり，ユーザが適切な条件設定をすることは難しい．

このような状況に対処するための有効な方法として，**適合フィードバック** [5] がある．適合フィードバックは，文書検索の分野で提案された，ユーザの検索要求を自動的に推定するための枠組みで，ユーザによる文書判定とその情報を利用した再検索を繰り返しながら，徐々に適合文書を集めていくという対話的アプローチを提供する．この方法を用いることで，ユーザが適合文書の判定さえすれば，適宜判別条件を自動修正していくことが可能となる．

我々はこれまで，再検索時に新たな適合文書を獲得するために有効な単語間の関係を学習する方法を提案し，新聞記事を使った検索実験を通して，その効果を確認している [9, 10]．本研究では，この方法を拡張し，WWW 検索エンジンが返すヒットリストから，適合ページのみを選びだすフィルタを自動生成するシステムを提案する．なお，このフィルタは複数のルール集合で構成され，各ルールは，キーワード，論理演算子，近接演算子，タグ情報の組み合わせによって表現される．一般に，Web ページはタグ付けがなされており，タグの種類によってページ内のテキストの重要度が異なる [8]．よって，検索範囲としてタグを指定することで，Web ページの特徴を活用することができ，より効率的な絞り込みが行えると考えられる．

以下の章では，まずこのシステムを使った検索過程について説明する．次に，システムの中心的な機能となる Web ページの構造を利用したルールの表現と生成方法について述べ，最後に，検索実験を行いシステムの能力を調べる．

2　適合フィードバックによる対話的 Web 検索

図 1 は，本研究で提案するシステムを使った検索処理過程の概要である．以下，各ステップで行われる手続きについて述べる．各手続きは，図中の番号の付いた矢印における処理と対応しており，ユーザ

側で行う操作とシステム側で行う操作の両方を記述している．

1. **初期検索**　初期条件として，検索エンジンに与える単語集合（以下**クエリ**と呼ぶ）と言語設定、日付指定等の入力をユーザに促し，入力された情報を検索エンジンに与え，検索結果を得る．

2. **ユーザによる検索結果の判定**　検索結果で上位にランクされたページからユーザに判定してもらい（通常上位 10 ページ程），適合ページ（**正例ページ**）と非適合ページ（**負例ページ**）に分けて，**訓練ページ**として保存する．

3. **訓練ページの解析**　フィルタを生成する際に必要な情報を訓練ページの解析により調べる．具体的には，各キーワードのページ中における出現場所（タイトルやアンカーテキストなど）と近接しているキーワードの組み合せを各訓練ページ毎に調べ，リテラルを生成して，フィルタを構成する条件候補集合を作る．

4. **フィルタの生成**　3 で得られた条件候補集合を使って，関係学習を行い，正例ページを含み負例ページを排除するフィルタを生成する．

5. **クエリの修正と再検索**　ページの判定や解析を行う中で，クエリに付け加えるべき単語などが見つかった場合，また適合ページが全く見つからない場合などに，クエリを修正して検索エンジンに与え，新たな検索結果を受け取る．

6. **フィルタリング処理**　検索結果で上位にランクされたページからフィルタリングを行い，フィルタを通過したページが必要数集まった時点で，その結果をユーザに提示する．ただし，既にユーザによって判定が行われたページは除く．

検索は以上の手順で進み，6 から 2 へ戻ることによりフィードバックが繰り返される．フィードバックを更に繰り返すかどうかは，そのときの検索結果を評価するユーザ側の判断で行うことができ，最終的に十分な情報が得られれば検索は終了となる．

以上の手続きの内，提示されるページの順位に直接影響する操作は，5 と 6 である．検索エンジンでは，これら 2 つの操作を支援するための機能を提供している場合が多い．5 のクエリの修正については，関連単語を選ぶための方法が，情報検索の分野においてこれまでに数多く研究されている [6, 7]．検索エンジンの中には，MetaCrawler[1] のように実際にクエリの単語に関連した語をいくつか提示してくれるものもある．6 のフィルタリングについては，オプションで複雑な論理式を指定することができるものなどはあるが，どのような設定をしたらいいのかを支援してくれる機能を提供するものは，今のところ存在しない．次章では，この 6 の操作を行うための Web ページの特徴を生かしたフィルタの表現と生成方法について詳しく述べる．

3　フィルタの表現と生成アルゴリズム

フィルタは複数のルールから構成され，各ルールはユーザから提示された正例ページと負例ページを訓練例とする分類学習を行うことにより得られる．本章では，まずルールの表現形式について述べ，次にその生成方法を示す．

3.1　ルールの表現

学習により獲得するルールは，キーワード，演算子，検索範囲の指定がなされたホーン節で表現する．ルールの条件部を構成するリテラルには，次のものを用いる．

- $ap(region\_type, word)$　：　ページ内の $region\_type$ 部分に $word$ が現れる．
- $near(region\_type, word1, word2)$ ：ページ内の $region\_type$ 部分で $word1$ と $word2$ が 10 単語以内に順不同で近接して現れる．

演算子は単語間の基本的な位置関係を表現する．近接関係は以前からその有効性が確認されていたが [3]，近年この関係を指定，または自動的に考慮する検索エンジンが増えている．また，Web 検索では，同じ単語でもページ内における出現場所によってその重要度が異なると考えられる．例えば，タイトルタグ内のテキストはそのページの主題を表現していることも多く，重要な手かがりとなる．よって

[1] http://www.metacrawler.com

ap, *near* リテラルともに *region_type* を加えることによって，より詳しい位置関係を指定している．*region_type* の種類は以下のものである．

- *title* : <TITLE>タグで囲まれたテキスト．
- *anchor* : <A>タグで囲まれたテキスト．
- *heading* : <H*n*> ($n = 1$〜4) タグで囲まれたテキスト．
- *para* : <P>タグで囲まれた 20 語以上からなるテキスト．

これらのリテラルにより，例えば次のようなルール集合が生成される．

$$\begin{cases} relevant \text{ :- } ap(title, \text{mobile}), ap(anchor, \text{PDA}). \\ relevant \text{ :- } near(para, \text{palm}, \text{os}). \end{cases}$$

各ルールは OR 関係にあり複数のルールの内一つでも満たせば適合ページと判定する．上のルール集合は，ページのタイトルに "mobile" が現れ，かつページ内に "PDA" が現れるアンカーテキストが存在するページ，またはページ内の同一段落で "palm" と "os" が近接して現れているページを表している．

入力：正例ページ集合 E^+，負例ページ集合 E^-，
　　条件候補リテラル集合 C，キーワード集合 K．
出力：ルール集合 R
変数：ルール *rule*,，除外リテラル l_1，
　　除外リテラル集合 S．
初期化：
　$K \leftarrow$ 検索式内の単語集合
　R, S, $l_1 \leftarrow$ *empty*
　rule $\leftarrow$ *relevant* :-
Repeat
　· *rule* を満たす正例ページ数 p と負例ページ数 n を調べる．
　if $n = 0$ **then**
　　· *rule* を R に加える．
　　· *rule* を満たす正例を E^+ から取り除く．
　　if E^+が空集合 **then** 終了
　　else *rule*, S, l_1 を初期化．
　else
　　· S 中のリテラルを除く C 中の全てのリテラルについて，重み付け情報利得 G を計算する．
　　if $G > 0$ となるリテラルがない **then**
　　　if *rule* のボディ部が空 **then**
　　　　· K にキーワードを一つ加える．
　　　　· C を新しく生成する．
　　　else
　　　　· S と *rule* を初期化する．
　　　　· l_1 を S に加え，l_1 を初期化
　　else
　　　· G が最大となるリテラルを l_{max} とする．
　　　if *rule* のボディ部が空 **then** $l_1 := l_{max}$
　　　· l_{max} を *rule* と S に加える．

図 2　ルール集合生成手続き

3.2　ルール集合の生成

フィルタとなるルール集合 R を生成するための手続きを図 2 に示す．この手続きは，Separate-and-Conquer 戦略 [1] を用いており，ルール（図 2 中の *rule*）を一つずつ生成し，R に追加する作業を繰り返す．*rule* が一つ生成されると，それによって被覆される文書が正例文書集合 E^+ から取り除かれるので，*rule* が生成される度に E^+ は減少していき，最終的に空集合となれば手続きが終了となる．同じ正例ページであっても文書中で使われる単語や，近接して現れる単語の組合わせが違うこともあり，そのページを識別するために有効な特徴は正例ページに依って異なる．

各ルールは空のボディ部にリテラルを一つずつ追加していき，負例を一つも含まなくなると完成となる．追加するリテラルは，条件候補リテラル集合 C の中から選ばれる．ここで C は，キーワード集合 K と *region_type* を引数に代入することにより作られる全てのリテラルの内，訓練ページで実際に成り立つものの集合を指す．具体的には次のようなリテラルである．

- K の全ての要素に関して，それを引数とした *ap* リテラルを各 *region_type* ごとに生成したものの中で，正例ページ内において成り立つもの．
- K の要素の全ての組み合わせに関して，それを引数とした *near* リテラルを各 *region_type* ごとに生成したものの中で，正例ページ内において成り立つもの．

また，追加するリテラルを選択する際の評価基準には，以下の式から計算される重み付き情報利得 G を用いる [4]．

$$G = e^{\oplus}_{new}\{I(e^{\oplus}_{old}, e^{\ominus}_{old}) - I(e^{\oplus}_{new}, e^{\ominus}_{new})\}$$

$$I(e^{\oplus}, e^{\ominus}) = -log_2 \frac{e^{\oplus}}{e^{\oplus} + e^{\ominus}}$$

$e^{\oplus}_{old}$, $e^{\ominus}_{old}$, $e^{\oplus}_{new}$, $e^{\ominus}_{new}$ はそれぞれ，リテラル追加前と追加後に満たす正例ページと負例ページの数である．これにより，正例ページ1つあたりの情報利得が大きく，かつ正例ページをより多く満たすリテラルが選ばれ，追加される．なお，G が最大となるリテラルが複数存在する場合，ランダムに選択する．

ルール生成途中では，全ての正例を満たすルール集合が生成されないまま，リテラルの選択候補がなくなり，探索が止まることがある．この時，$rule$ のボディ部にリテラルが1つ以上追加された状態であれば，その時点で $rule$ に追加されているリテラルを全て破棄し，$rule$ の生成をやり直す．その際，同じ探索を繰り返さないため，生成をやり直す前の $rule$ に追加されたリテラルのうち，最初に追加されたもの（図2中の変数 l_1）を予め除外しておく．そうでない場合，つまり $rule$ のボディ部が空の状態である場合，K にキーワードを新たに加え，C を新しく生成することによって選択可能なリテラルを作る．

追加するキーワードは，正例ページ集合 E^+ から選ぶ．まず各ページ中の<P>タグで囲まれた20単語以上からなる段落の内，クエリ内の単語を少なくとも1つ含むもののみを集め，これを T とする．次に T に出現する全単語集合 W の各要素 w_i に対して，以下の式から重要度を計算する．

（w_i の重要度） ＝
（T 中における w_i 平均出現頻度）×
（w_i が現れる T 中の段落数）

この重要度が最も高いもので，クエリに使われておらず，まだ追加されてないものを新しく追加する．

4 実験

提案システムの有効性を調べるために，2章で説明した手順に従った検索実験を行った．

4.1 実験方法

検索エンジンには，検索精度が良いとされるGoogle[2]を用い，英語で記述されたページを検索対象とした．検索要求には，TREC[3]の small web track で用いられたトピックから20個 (401～420) を選んだ．図3は実験で用いたトピックの例で，検索要求や適合文書とする際の判定基準などが記述されている．適合ページの判定は，これらの記述に従い，全て同一人物が行った．各トピックの<title>タグには，1～3単語が記されており，これを初期クエリとして与えた．また，この実験は，システムが生成するフィルタの性能評価を目的とするため，2章で述べた手続き5の処理は行わない．

評価方法として，Google のみを使った通常のWWW検索とフィルタを用いた提案システムによる検索について，一定数のページを見た場合に得られた適合ページ数をトピック別に比較した．どちらの検索も，各トピックにつき合計50ページを調べた．Google のみによる検索では，初期検索結果の上位50ページを調べた．提案システムを用いた検索では，10ページ判定する毎にフィルタを新しく生成し，ヒットリストの上位からまだ調べていないページについてフィルタリングを行い，10ページ得られた時点で判定を行う操作を4回繰り返した．

<num> Number: 401
<title> foreign minorities, Germany

<desc> Description:
What language and cultural differences impede the integration of foreign minorities in Germany?

<narr> Narrative:
A relevant document will focus on the causes of the lack of integration in a significant way; that is, the mere mention of immigration difficulties is not relevant. Documents that discuss immigration problems unrelated to Germany are also not relevant.

図3 トピックの例

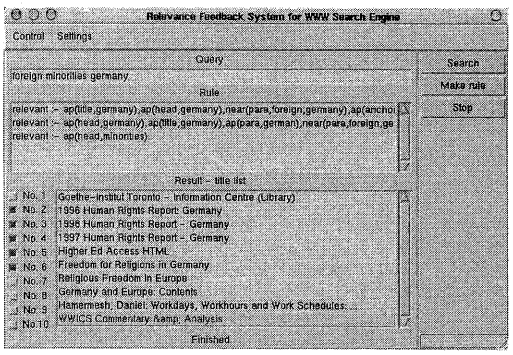

図4 システムインタフェース

[2]http://www.google.com
[3]http://trec.nist.gov

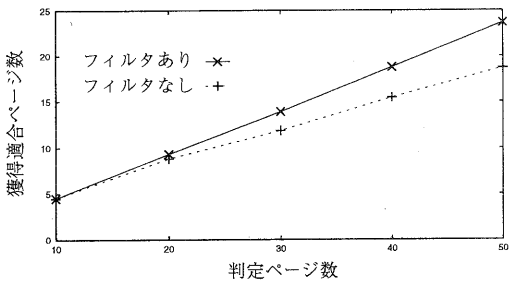

図 5　獲得適合ページの平均値

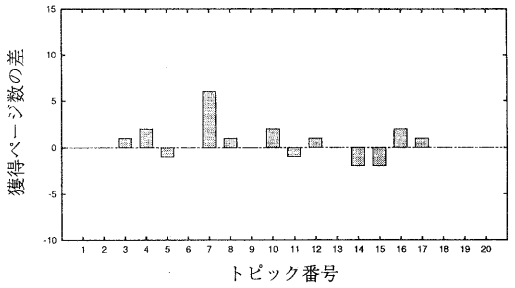

図 6　トピック別獲得適合ページ数（20 ページ判定後）

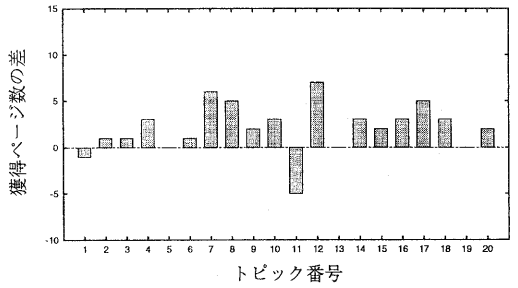

図 7　トピック別獲得適合ページ数（30 ページ判定後）

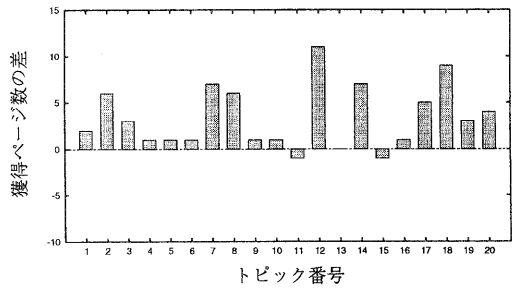

図 8　トピック別獲得適合ページ数（40 ページ判定後）

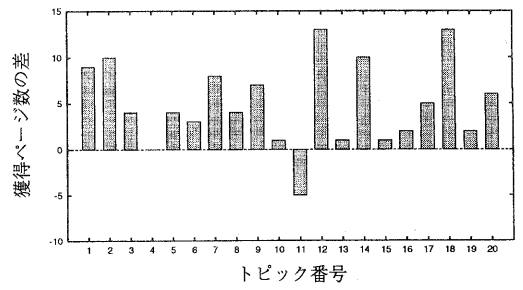

図 9　トピック別獲得適合ページ数（50 ページ判定後）

図 4 は，提案システムのインタフェースである．“Query” 部分に入力された単語が検索エンジンに入力され，返ってきたヒットリストをフィルタリングしたものの上位 10 ページのタイトルが “Result” 部分に表示される．タイトルをクリックすると，ブラウザが立ち上がりそのページが閲覧できるようになっている．ページをみた後，タイトル番号横のボタンを押すことでそのページが適合ページであることをマークする．マークがないものは全て非適合ページとして扱われる．このあと “Make Rule” ボタンを押すことでフィルタルールが生成され，“Rule” 部分に表示される．

4.2　実験結果

図 5 は，提案システムを使った検索（フィルタあり）と使わない検索（フィルタなし）を行った際の，判定ページ数と獲得適合ページ数の関係を示したものである．各値は，20 個のトピックについての平均値である．最初の 10 ページはどちらも同じものを評価するので差はないが，それ以降提案システムを用いたものは，フィルタリングによって提示されるページが違うため，差が生じている．50 ページを判定した段階でフィルタリングを行ったものは平均 5 ページ多く適合ページが得られている．このように平均的に見てフィルタリングの効果が現れていると言えるが，トピックによってその効果は大きく異なる．

図 6～図 9 は獲得適合ページ数の差 D をトピック別に示したものである．提案システムを用いた場合の適合ページ数を A，用いない場合を B とすると，$D = A - B$ で示される．図 6 では，判定ページ数が少ないので正例ページをあまり得られないトピッ

クが多くあること，また検索エンジンのヒット率もまだ高いこと等が原因でフィルタリングの効果はほとんど見られず，逆効果が現れているトピックも4つある．図7では多くのトピックで効果が現れ始めている．図8では，図7で効果の出たトピックの多くで差が広がっている．図9では効果が得られているものとそうでないものとの差がはっきり分かるようになる．効果の現れ方はトピックによってまちまちであり，11，13，15番のトピックのように差が現れないものや負の値となるものがあるものの，判定ページの増加と共にほとんどのトピックについて差が正の値となっており，全体的に見て，フィルタリングの効果が現れているといえる．

5 考察

実験結果から，ほとんどのトピックにおいて，獲得適合ページ数の差が正の値になっているが，約半数のトピックについては，検索エンジンのみでも十分な適合ページが得られるため目立った効果が現れなかった．ここでは，うまく効果が現れたものとそうでないものの例を示す．

12番と18番のトピックは，効果が一番良く現れている．12番の検索要求は，「空港における有効なセキュリティシステムにはどのようなものがあるか」というものである．検索エンジンがこの問に関して返すヒットリストのなかには，適合ページの他に，「旅行者が心得ておくべき注意点」を紹介しているページが多く含まれた．フィルタリングを使った検索では，これらの非適合ページを多く排除できたため効果が大きく現れた．表1は，このトピックに関する検索で生成されたルールの内，適合ページを多く獲得したものの一部である（各ルールともボディ部のみを示してある）．各ルールをみてわかるように，“airport” と “security” を基本に，その他の補助単語を組み合わせることによって効果的なフィルタリングが行われていた．

また，18番の検索要求は，「キルト製品でどのようにして収入が得られているか」というもので，キルトについて書かれた本，キルト教室等を紹介したものが適合ページとなる．フィルタリングを使った検索では，これらのページの特徴をうまく捕らえて効果を上げており，特にキルト製品をオンライン販売しているページが多く検索されていた．表2に，同じく適合ページを多く獲得したルールの一部を示す．このフィルタでは，“online” とその他の単語が組み合わされることにより，効果的なフィルタリングが行われていた．

表1　トピック12番で生成されたルールの例

```
:- ap(anchor,screening).
:- near(para,security,system),ap(title,airport).
:- near(para,security,airports),
   near(para,security,access).
:- near(para,security,airports),
   near(para,faa,system).
```

表2　トピック18番で生成されたルールの例

```
:- ap(para,online),ap(title,quilts).
:- ap(anchor,online),ap(title,quilting),
   ap(anchor,quilting).
:- ap(para,block),near(para,quilt,block),
   ap(anchor,fabric).
:- ap(title,quilting),ap(anchor,fabric).
```

表3　トピック11番で生成されたルールの例

```
:- ap(anchor,shipwreck).
:- ap(anchor,shipwreck),ap(anchor,salvaging).
```

逆に効果が全く見られなかった11番のトピックは，「難破船を引き上げ宝物を得るために必要な情報」が得られるページを探すというものである．このトピックに適合するページは，リンク集，掲示板，ニュース，宝探しのグループのホームページなどの多種類のページが少しずつあるため，有効なフィルタ生成が行われず，効果が出なかったと考えられる．表3には，生成されたルールの中で適合ページが得られたものを示してある．これらを見てわかるように，生成されたルール数が少なく，条件部もWebページを絞り込むには不十分であるため，フィルタリングの効果が現れなかった．

6 まとめ

本研究では，検索エンジンが返すヒットリストを逐次的にフィルタリングすることによって，適合ページを効率よく選別する対話的な検索処理システムの構成とその検索手続きについて説明した．提案したシステムは，Webページ中のキーワードの位置関

係と構造的条件を加味してフィルタを自動生成し，複雑な絞り込みを行うことができる．本研究では，これを実験を通して確認することができた．

現行の検索エンジンでは，このようにユーザからのフィードバック情報を処理し，ユーザ個別の情報検索を支援する枠組みはまだ提供されておらず，検索エンジンをより有効に活用するために本研究で述べたアプローチは十分有効であると考えられる．

今後の課題としては，ユーザがページの判定をスムーズに行うためのインタフェースの工夫や視覚化機能を追加することである．また，適合フィードバックを行う際にユーザの負担となる判定ページの必要数をできるだけ減らすことは重要であり，クラスタリング機能などの追加を検討している．

参考文献

[1] Furnkranz, J.: Separate-and-Conquer Rule Learning, *Artificial Intelligence Review*, Vol.13, No.1 (1999)

[2] Jansen, M.B.J., Spink, A., Bateman, J. and Saracevic, T.: Real life information retrieval: A study of user queries on the web, *SIGIR Forum*, Vol.32, No.1, pp.5-17 (1998)

[3] Keen, E.M.: Some aspects of proximity searching in text retrieval system, *Journal of Information Science*, Vol.18, No.2, pp.89-98 (1992)

[4] Quinlan, J.R., and Cameron-Jones, R.M.: Induction of Logic Programs: FOIL and Related Systems, *New Generation Computing*, Vol.13, Nos.3,4, pp.287-312 (1995)

[5] 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)

[6] Mitra, M., Singhal, A., and Buckley, C.: Improving automatic query expansion, In *Proceedings of the 21st annual international ACM SIGIR*, pp.206-214 (1998)

[7] Xu, J. and Croft, W.B.: Query expansion using local and global document analysis, In *Proceedings of the 19th annual international ACM SIGIR*, pp.4-11 (1996)

[8] Zhang, D. and Yisheng, D.: An efficient algorithm to rank Web resources, *WWW9*, pp.449-455 (2000)

[9] 岡部，山田：関係学習を用いた対話的文書検索，人工知能学会誌，Vol.16, No.1P，(2001)

[10] Okabe, M. and Yamada, S.: Interactive Document Retrieval with Relational Learning, In *Proceedings of the 16th ACM Symposium on Applied Computing*, to appear