フィルタリングルールの逐次的学習による対話的 Web ページ検索*

岡部 正幸[†]・山田 誠二[‡]

Interactive Web Page Retrieval by Iterative Learning of Filtering Rules*

Masayuki OKABE[†] and Seiji YAMADA[‡]

Web search engines are useful tools which can meet the various information needs of users. However they often return hit-lists which contain many unnecessary pages. This paper proposes a method which automatically removes those unnecessary pages by learning filters through relevance feedback. Filters consist of several rules, each of which describes conditions for discriminating relevant pages using useful cooccurrences or proximities among words, and parts in a page where those conditions are applicable. Thus they enable to classify a variety of pages precisely. Moreover, users are free to generate and apply filters at anytime. Through experiments we demonstrate that our filters increase the number of relevant pages we can get in a retrieval, compared to representatives of web search engine and relevance feedback method.

1. はじめに

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が、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 語と少ないため [1], 検索目的とは関係のない多くのページとともに数千もの Web ページがヒットしてしまうことなどがよくある。また、ヒットリストの上位にユーザの要求を満たす Web ページ（適合ページとよぶ）が集中しているとは限らず、順位が低くても、適合ページがたくさん見つかる場合も多い。効率的な検索を行うには、ユーザの検索意図を反映したランキングを行うことが必要であるが、ランキング方法はそれぞれの検索エンジン

が独自に行っており、多くの場合その設定をユーザが調節することはできない。そこで、本研究では、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自動的に選別するフィルタリング処理を行うことによって検索を効率化する。

しかし、一般にユーザは検索を始める際に、目的とする情報を含む 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 ページの内容が目的に沿うものであるかどうか、つまり適合ページであるか否かを判断することはできても、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（フィルタ生成）を行うことは難しい。本研究では、このフィルタ生成をユーザにできるだけ負担をかけることなく行うために、適合フィードバック [2] を用いた対話型の検索システムを提案する。適合フィードバックは、文書検索の分野で提案された、ユーザの検索要求表現（クエリ）を自動的に修正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

* 原稿受付 2002 年 11 月 11 日

[†] 豊橋技術科学大学 マルチメディアセンター Multimedia Center, Toyohashi University of Technology; Tenpakucho, Toyohashi city, Aichi 441-8580, JAPAN

[‡] 国立情報学研究所 National Institute of Informatics; 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, JAPAN
Key Words: web page filtering, relevance feedback, rule learning.

ている [3,4]. 本研究では, この方法を Web 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の自動生成へ適用する. 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, タグ情報の組合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 [5]. 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.

このように, ユーザから求めている情報を例示してもらうことによって, より精度の高い Web ページ収集を行うことを目的としたシステムは, これまでにいくつか提案されている. Syskill & Webert[6] は, 情報利得を使ったキーワード抽出とベイズ分類器によるフィルタリング機能を持った Web ページ収集システムである. 現在見ているページが持つリンクページの中から, もしくは検索エンジンに直接クエリを与えることにより, ユーザが興味を持つと思われるものを推薦する. このシステムはキーワードのみを特徴として用いており, 本研究のようにキーワード間の関係やタグ情報などは考慮していない. また, 検索エンジンが返すヒットリストのフィルタリングは行わず, そのまま用いている. Focused Crawler[7] は, フィルタリング機能を持った Web ページ収集ロボットである. このシステムは, 通常の検索ロボットのように任意のページを収集するのではなく, 特定の検索要求に見合う Web ページのみを選択的に収集する. たどるべきリンクを決定するために, ユーザより与えられる訓練例を使ったベイズ分類学習を行う. 学習はページ内に現れる単語の頻度に基づいて行われるため, キーワード間の関係やタグ情報は考慮されない. また, あらかじめ与えられた抽象的な分類クラスをターゲットとしており, 本研究のように任意の検索要求には対応しないシステムといえる.

以下の章では, まず 2 章でフィルタリングを伴う検索過程について説明する. 次に 3 章で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フィルタリングルールの表現と生成方法について述べ, 4 章で検索実験を行いシステムの能力を調べる. 5 章ではシステムの効果をより詳しく考察し, 最後に 6 章で本研究をまとめる.

2. 適合フィードバックによる対話的 Web 検索

Fig. 1 は, 本研究で提案するシステムを使った場合の検索処理の概要である. 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 (Initial search)**: 初期条件として, 検索エンジンに与える単語集合 (以下クエリとよぶ) 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 (User feedback)**: 検索結果で上位にランクされた Web ページ (通常上位 10 ページ程) をユーザに判定してもらい, 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 (Analysis of training pages)**: 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得る. 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合せを各訓練ページごとに調べ, 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 (Filter generation)**: (3) で得られた条件候補集合を使って, 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 (Query expansion and retry)**: 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¹.
- (6) **フィルタリング処理 (Filtering)**: 検索結果で上位にランクされた Web 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, すで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.

以上の手続きのうち, 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, これら二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 [8,9]. 検索エンジンの中には, MetaCrawler²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

¹本論文では, この処理に関して新たな提案は行っていない. また, この処理を含めた実験も行っていないため, 図中では点線にしている.

²<http://www.metacrawler.com>

Fig. 1 Information retrieval with filtering

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おり[10]、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。たとえば、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が多く、重要な手がかりとなる。よって 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。

- $title$: $\langle TITLE \rangle$ タグで囲まれたテキスト。
- $anchor$: $\langle A \rangle$ タグで囲まれたテキスト。
- $heading$: $\langle Hn \rangle$ ($n=1\sim 4$) タグで囲まれたテキスト。
- $para$: $\langle P \rangle$ 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、たとえば次のようなルール集合が生成される。

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

各ルールはOR関係にあり、複数のルールのうち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。

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Fig.2に示す。

3.2.1 Separate-and-Conquer 戦略

この手続きは、Separate-and-Conquer 戦略[11]を用いており、ルール (Fig.2の $rule$) を一つずつ生成し、 R

Input : Set of positive pages E^+ ,
Set of negative pages E^- ,
Set of condition literals C ,
Keyword set K

Output : Rule set R

Variable : $rule$ (for a rule),
 l_1 (for a literal to remove),
 S (for a set of literals to remove)

Initialize :
 $K \leftarrow \text{keywords in a query}$
 $R, S, l_1 \leftarrow \text{empty}$
 $rule \leftarrow \text{relevant} :-$

Repeat

- Count the number of positive(negative) pages $p(n)$ which satisfy(do not satisfy) $rule$.
- if** $n=0$ **then**
 - Add $rule$ to R .
 - Remove positive pages which satisfy $rule$ from E^+ .
- if** E^+ is empty **then exit**
- else** Initialize $rule, S, l_1$.
- else**
 - Calculate weighted information gain G for all literals in C without literals in S .
 - if** There is no literal which satisfies $G > 0$ **then**
 - if** Body of $rule$ is empty **then**
 - Add a keyword to K .
 - Generate a new C .
 - else**
 - Initialize S and $rule$.
 - Add l_1 to S
 - Initialize l_1 .
 - else**
 - Let l_{max} be a literal which has largest G .
 - if** Body of $rule$ is empty **then** $l_1 := l_{max}$
 - Add l_{max} and $rule$ to S .

Fig. 2 Process of rule generation

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、 $rule$ が生成されるたびに 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せが違ふこともあり、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よって異なる。

3.2.2 重み付き情報利得を用いたルール生成

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すべてのリテラル

のうち、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りテラルである。

- K のすべての要素を引数 $word$ に代入した ap リテラルを各 $region.type$ ごとに生成したものの中で、少なくとも一つの正例ページで成り立つもの。
- K の要素のすべてのペアを引数 $word1, word2$ に代入した $near$ リテラルを各 $region.type$ ごとに生成したものの中で、少なくとも一つの正例ページで成り立つもの。

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 [12].

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

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

$e_{old}^{\oplus}, e_{old}^{\ominus}, e_{new}^{\oplus}, e_{new}^{\ominus}$ 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、正例ページ一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。

3.2.3 バックトラックとキーワードの追加

ルール生成途中では、すべ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。このとき、 $rule$ のボディ部にリテラルが一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すべて破棄し、 $rule$ 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 (Fig. 2 中の変数 l_1) をあらかじめ除外しておく。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 $\langle P \rangle$ タグで囲まれた 20 単語以上からなる段落のうち、クエリ内の単語を少なくとも一つ含むものののみを集め、これを T とする。つぎに 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。

$$(w_i \text{ の重要度}) = (T \text{ 中における } w_i \text{ 平均出現頻度}) \times (w_i \text{ が現れる } T \text{ 中の段落数})$$

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

提案手法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

提案手法を検索エンジンに適用する利点とこれまでの典型的な適合フィードバック手法に対する性能比を調べるため、次の三つのタイプの検索についてそれぞれ50ページを判定し、得られた適合ページ数を比較した。

- (1) 検索エンジンのみを使った検索 (engine と表記)
- (2) ベクトル空間モデルを用いた適合フィードバックを行った検索 (vector と表記)
- (3) フィルタリングルールを用いた適合フィードバックを行った検索 (rule と表記)

検索実験を行うために必要なユーザの検索要求として、コンテスト形式による検索システムの性能評価を目的とした会議である TREC¹ の Small Web Track において利用された検索課題 (トピック) から初めの20個 (No.401~420) を選んで用いた。このトピックの選択に意味は全くランダムな選択と等しい。

適合ページの判定は、著者1人を含む4人の被験者が行った (日本人男性1人、ロシア人男性2人、ブルガリア人男性1人、うち3人は大学勤務、1人は博士課程学生。全員、十分な英語の読解能力を持つ)。各トピックに対して初期クエリ (各トピックの <title> タグに記された1~3個の単語、Table 2 参照) を検索エンジンに入力することによって得られたヒットリストの上位50ページをダウンロードしておき、各ページに対してその内容がトピックの要求に適合するかどうかをあらかじめ各被験者に判定してもらった。よって三つの検索はこの共通する50ページに対して行われた。また、被験者には特定の手法の検索結果という情報が与えられないため、判定を恣意的に行うことはできない。以下 (1)~(3) での具体的な検索方法と判定手続きについて説明する。

(1) の検索では、検索エンジンに、検索精度が良いとされる Google² (英語版) を用い、初期検索で得られたヒットリストの上位50ページを調べた。

(2) の検索のベクトル空間モデルを用いた適合フィードバックは、2章で述べた手続き (5) のように検索エンジンに与えるクエリを直接修正する方法ではないが、クエリベクトルに含まれる単語の重みを修正することがクエリの修正と対応しており、2章の手続き (5) の代替方法と見なすことができる。一般的に用いられるクエリベクトルの修正式を次式に示す。

$$Q^{new} = \alpha Q^{old} + \beta \frac{1}{N^+} \sum_{rel} D^+ - \gamma \frac{1}{N^-} \sum_{non\ rel} D^-$$

D^+ , D^- はそれぞれユーザによって示された適合ページと非適合ページから作られる単語ベクトル, N^+ , N^- はそれらの数を表す。 α, β, γ のパラメータはそれぞれ8,16,4の組合せで使われることが多く、この実験でもこの組合せを用いた [13]。ページ内の単語の重み付けは TREC コンテストで高い性能を示した okapi システムが採用している方法を用いた [14]。重み v_i の計算式を次式に示す。

$$v_i = \frac{tf}{tf + 0.5 + 1.5 \frac{doclen}{avgdoclen}} \cdot \frac{\log\left(\frac{colsize + 0.5}{docf}\right)}{\log(colsize + 1)}$$

tf はページ内での単語の頻度, $docf$ は単語が現れるページ数, $doclen$ はページ内に現れる単語数, $avgdoclen$ は500ページでの $doclen$ の平均値, $colsize$ は総ページ数 (この実験では500) を表している。ヒットリストの上位10ページを判定するごとにクエリベクトルの更新を行い、この更新されたクエリベクトルを使って500ページを適合度の高い順にソートし、新しいヒットリストを作る。この作業を4回繰り返すことによって合計50ページの判定を行った。

(3) の検索は初期検索で得られたヒットリストの上位から10ページ判定するごとにフィルタを新しく生成し、ヒットリストはそのままフィルタのルールを満たすかどうかのチェックを1位にランクされたページからやり直す。これによりフィードバック前にフィルタを通過しなかったページが新たに適合ページとなることもある。実験ではこの作業を4回繰り返し、合計50ページの判定を行った。

実験に先立ち、2章で説明した検索を実際の実験エンジンを使って行うことのできるインタフェースを作成した。Fig.3は、そのインタフェースの概観である。クエリ入力に加え、適合性判定のマーキング、ルール生成などを行うことができる。

Fig. 3 Interface

¹<http://trec.nist.gov>

²<http://www.google.com>

Fig. 4 Number of relevant pages averagedly acquired

Fig. 5 Number of relevant pages acquired for each topic

Table 1 Result of *t*-test

	engine vs. rule	vector vs. rule	engine vs. vector
subject A	15.9 < 19.6 (S)	19.7 > 19.6 (S)	15.9 < 19.7 (S)
subject B	12.5 < 14.6 (NS)	10.2 < 14.6 (S)	12.5 > 10.2 (NS)
subject C	16.3 < 19.0 (S)	16.2 < 19.0 (S)	16.3 > 16.2 (NS)
subject D	16.1 < 19.8 (S)	16.5 < 19.8 (S)	16.1 < 16.5 (NS)
Total	60.8 < 73.0 (S)	62.6 < 73.0 (S)	60.8 < 62.6 (NS)

S: Significant

NS: Not significant

4.2 実験結果

Fig. 4 は、前節で説明した三つの検索結果から得られた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 4 人の被験者と 20 個のトピックについての平均値であり、1 人が 1 トピックについて一定数の判定を行った場合に得られる適合ページ数の期待値を表している。最初の 10 ページはどの検索でも同じものを評価するので差はないが、それ以降判定ページが増えるにつれて提案手法による検索と他の二つの検索との差が開いている。50 ページを判定した段階で平均 3 ページ前後多く適合ページが得られている。この差が有意なものであるかどうかを調べるため、50 ページの判定後に、各被験者の各トピックで得られた適合ページ数の分布の平均値の差を検定した。検定方法には student の *t* 検定を用いた（有意水準 5% の両側検定）。また被験者全員の合計値（各トピックにつき被験者全員の適合ページ数を合計したもの）の分布についても検定を行った。Table 1 にその結果を示す。各被験者ごとに三つの検索間での平均値の大小の比較とその検定結果が示されている。Table 1 の中の S, NS の各記号はそれぞれ「有意である」(Significant), 「有意でない」(Not Significant) ことを示している。被験者個別で違いはあるものの、フィルタリングを用いた検索は他の二つの検索と比べておおむね有意な差が得られていると見ることができる。以上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いえるが、トピックによってその効果は大きく異なる。Fig. 5 は 50 ページ判定後の獲得適合ページ数をト

ピック別に示したものである。各値は 4 人の被験者の平均値であり、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と現れている。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が、効果の現れ方はトピックによって様々である。次章では、効果が現れたトピックとそうでないものについていくつか取り上げ考察する。

5. 考察

実験結果から、提案手法による検索結果が全体的に優位であることがわかったが、Fig. 5 に見られるようにトピックによってその効果にはばらつきがある。ここでは提案手法と他の二つの方法による検索結果を比較しながら、提案手法の特徴を考察する。

5.1 検索エンジンとの比較

まず検索エンジンとの比較では、12 番と 18 番のトピックに提案手法によるフィルタリングの効果がよく現れている。Table 2 はトピック番号 12 番の検索課題である。この検索課題に対して検索エンジンが返すヒットリストの中には、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ていた。フィルタリングを使った検索では、このような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。Table 3 は、このトピックに関する検索過程で生成されたルールのうち、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

Table 2 Topic no. 12

<num> Number: 412
<title> airport security
<desc> Description: What security measures are in effect or are proposed to go into effect in airports?
<narr> Narrative: A relevant document could identify a specific airport and describe the security measures already in effect or proposed for use at that airport. Relevant items could also describe a failure of security that was cited as a contributing cause of a tragedy which came to pass or which was later averted. Comparisons between and among airports based on the effectiveness of the security of each are also relevant.

Table 3 Example of rules generated for topic no. 12

$\text{:- ap(anchor,screening).}$
$\text{:- near(para,security,system),ap(title,airport).}$
$\text{:- near(para,security,airports),near(para,security,access).}$
$\text{:- near(para,security,airports),near(para,faa,system).}$

基本に, “faa”(The federal Aviation Administrationの略称)や“screening”といった具体的な機関名やシステム名を意味する単語が組み合わされることにより, 精度の高いフィルタリングが行われていた. また, Table 4は18番の検索課題である. この検索課題に対しては, 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が, 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. Table 5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. この中で, “online”と“quilt”の二つの単語が用いられているルールにより, オンラインショップ関連のページが選び出されていた. また, “fabric”と“quilt”が用いられているルールで得られたページでは, 織物コレクションの一部としてキルトが紹介されていた.

上の二つのトピックは, より多くの適合ページを見つけ出すという効果は同じものの, 得られた適合ページは必ずしも検索エンジンが返すヒットリストの上位から順に選ばれているわけではない. これは次のような(擬似的な)再現率を計算することによって示される. つまり, フィルタを使って50ページ調べた場合に得られる適合ページ集合を R_1 , 検索エンジンが返すヒットリストの上位50ページに含まれる適合ページ集合を R_2 とした場合に $(R_1 \cap R_2)/R_1$ で示される値はヒットリストに元々含まれる適合ページをフィルタが残している割合を示している. 12番のトピックではこの値が0.71と数値が高く, 上位50ページに含まれる適合ページの多くが検索

Table 4 Topic no. 18

<num> Number: 418
<title> quilts, income
<desc> Description: In what ways have quilts been used to generate income?
<narr> Narrative: Documents mentioning quilting books, quilting classes, quilted objects, and museum exhibits of quilts are all relevant. Documents that discuss AIDs quilts are irrelevant, unless there is specific mention that the quilts are being used for fundraising.

Table 5 Example of rules generated for topic no. 18

$\text{:- ap(para,online), ap(title,quilts).}$
$\text{:- ap(anchor,online), ap(title,quilting),}$ $\text{ap(anchor,quilting).}$
$\text{:- ap(para,block), near(para,quilt,block),}$ $\text{ap(anchor,fabric).}$
$\text{:- ap(title,quilting), ap(anchor,fabric).}$

されているのに対し, 18番のトピックでは0.47と低い数値となっており, 50位よりも下位にランクされている適合ページも多く検索されていることを示している. 18番のトピックに関しては, 先に述べたように同種の適合ページ(この場合オンラインショッピングサイト)に特化されたフィルタが生成されていることによると考えられるが, 一般に, 生成されるフィルタの質は生成時に用いる訓練ページの質に依存する. 本実験ではトピックごとに記述された客観的基準以外に適合ページの質を問わないため, 判定したページすべてを訓練ページとして与えているが, 実際の検索ではユーザは検索過程を通して要求する適合ページの質を決めていくことも多いため, そのような場合には求める適合ページの質を考慮した訓練ページの与え方が必要となる.

5.2 ベクトル空間モデルとの比較

次にベクトル空間モデルによる適合フィードバックを用いた検索と比較すると, 6番のトピックにフィルタリングの効果がよく現れている. このトピックは, Table 6の検索課題に見られるようにパーキンソン病の治療方法を探すものであるが, この病気の治療法は薬による処方为主であるため, クエリベクトル中では, “agonist”, “levodopamin”, “selegiline”など具体的な薬を示す単語が高い重みを持つ. しかし, 肝心の“parkinson”や“treatment”などの単語の重みがそれほど高くないため, アルツハイマー病について書かれたページや単に薬の説明がなされたページなどパーキンソン病との関連について少し触れられた程度のページが検索されていた. 一方, 提案手法の方はTable 7に見られるように薬の名前を表す単語はほとんど使われておらず, “parkinson”,

Table 6 Topic no. 6

<num> Number: 406
<title> Parkinson's disease
<desc> Description: What is being done to treat the symptoms of Parkinson's disease and keep the patient functional as long as possible?
<narr> Narrative: A relevant document identifies a drug or treatment program utilized in patient care and provides an indication of success or failure.

Table 7 Example of rules generated for topic no. 6

<code>:- near(head, parkinson, disease).</code>
<code>:-near(para,disease,patients),</code>
<code> near(para,parkinson,treatment),</code>
<code> ap(head,parkinson).</code>
<code>:- near(para,disease,patients),</code>
<code> near(anchor,parkinson, treatment).</code>
<code>:- ap(anchor,disease), ap(para,patients),</code>
<code> ap(title,parkinson), near(anchor,parkinson,disease).</code>
<code>:- near(para,patients,levodopa).</code>

“treatment”などの単語の出現位置や組合せによってフィルタを構成することで適合ページの選別を行っていた。

5.3 効果が現れない例

多くのトピックに関して良好な結果を残す半面、11番のトピックなど、他の二つの手法に対して、効果が全く見られなかったトピックもあった。Table 8はこのトピックの検索課題である。このトピックに適合するページとしては、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が挙げられるが、それぞれの適合ページは少しずつしかないので、本論文で説明したルール生成アルゴリズムでは、個々のページに特化された特徴が生成されてしまうことから、類似ページの選別が十分行えず、効果が出なかったと考えられる。Table 9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. おわりに

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方法とフィルタを構成するルールの生成アルゴリズムについて説明した。提案手法は、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

Table 8 Topic no. 11

<num> Number: 411
<title> salvaging, shipwreck, treasure
<desc> Description: Find information on shipwreck salvaging: the recovery or attempted recovery of treasure from sunken ships.
<narr> Narrative: A relevant document will provide information on the actual locating and recovery of treasure; on the technology which makes possible the discovery, location and investigation of wreckages which contain or are suspected of containing treasure; or on the disposition of the recovered treasure.

Table 9 Example of rules generated for topic no. 11

<code>:- ap(anchor,shipwreck).</code>
<code>:- ap(anchor,shipwreck),ap(anchor,salvaging).</code>

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、ユーザが持つ情報要求の具体化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などが挙げられる。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の追加なども必要である。

参考文献

- [1] M. B. Jansen, A. Spink, J. Bateman and T. Saracevic: Real life information retrieval: A study of user queries on the web; *SIGIR Forum*, Vol. 32, No. 1, pp. 5-17 (1998)
- [2] G. Salton and C. Buckley: Improving retrieval performance by relevance feedback; *Journal of the American Society for Information Science*, Vol. 41, No. 4, pp. 288-297 (1990)
- [3] 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. 1, pp. 139-146 (2001)
- [4] M. Okabe and S. Yamada: Interactive document retrieval with relational learning; *Proc. 16th ACM Symposium on Applied Computing*, pp. 27-31 (2001)
- [5] D. Zhang and D. Yisheng: An efficient algorithm to rank Web resources; *Proc 9th Int. World Wide Web Conf.*, pp. 449-455 (2000)
- [6] M. Pazzani, J. Muramatsu and D. Billsus: Syskill & Webert: Identifying interesting web sites; *Proc. AAAI* (1996)
- [7] S. Chakrabarti, M. Berg and B. Dom: Focused crawling: A new approach to topic-specific web resource

- discovery; *Proc 8th Int. World Wide Web Conf.* (1999)
- [8] M. Mitra, A. Singhal and C. Buckley: Improving automatic query expansion; *Proc. 21st annual international ACM SIGIR*, pp. 206-214 (1998)
- [9] J. Xu and W. B. Croft: Query expansion using local and global document analysis; *Proc. 19th annual international ACM SIGIR*, pp. 4-11 (1996)
- [10] E. M. Keen: Some aspects of proximity searching in text retrieval system; *Journal of Information Science*, Vol. 18, No. 2, pp. 89-98 (1992)
- [11] J. Furnkranz: Separate-and-conquer rule learning; *Artificial Intelligence Review*, Vol. 13, No. 1 (1999)
- [12] J. R. Quinlan and R. M. Cameron-Jones: Induction of logic programs: FOIL and related systems; *New Generation Computing*, Vol. 13, Nos. 3,4, pp. 287-312 (1995)
- [13] M. Iwayama: Relevance feedback with a small number of relevance judgements: Incremental relevance feedback vs. document clustering; *Proc. 20th Annual International ACM SIGIR*, pp. 10-16 (2000)
- [14] A. Leuski and J. Allan: Improving realism of topic tracking evaluation; *Proc. 22th Annual International ACM SIGIR*, pp. 89-96 (2002)
-