2005－DBS－137（II）（45）
2005／7／14

例示データに基づく選択的ウェブクローリング手法について

張建偉[†]，　石川佳治[†,††]，　黒川沙弓[†]，　北川博之[†,††]

本稿では，ユーザが提供する例示データに基づく選択的な情報収集手法の提案を行う．本手法の特徴の一つは，ウェブページを動的に収集しつつ，収集したウェブページから情報抽出を行い，ユーザに提供された例示データを拡充する点である．また，拡充されたデータからなるデータベースとウェブを融合し，効率よく選択的クローリングを図る点も特徴である．情報抽出およびクローリングは，ユーザからのフィードバックに応じて適応的に処理される．本手法は内容解析，リンク解析とトピック主導型クローリングの技術を統合することで，例示データに関連する情報を保持するページの効果的な収集の実現を図る．

A Selective Web Crawling Method Based on User Examples

Jianwei Zhang[†]，　Yoshiharu Ishikawa[†,††]，　Sayumi Kurokawa[†]，　Hiroyuki Kitagawa[†,††]

In this paper, we propose a selective web crawling method to collect web pages based on example records provided by a user. One of the features of our method is that example records are expanded dynamically with additional records extracted from the crawled HTML pages. Moreover, our system integrates the database composed of example and additional records and the web to achieve an efficient and selective crawling. Information extraction processing and crawling processing are processed adaptively according to the feedbacks from the user. Our method combines the tecniques of contents analysis, link analysis and topic-focused crawling. Therefore, the method will lead an efficient collection of web pages which contain information related to example records.

1　はじめに

ウェブの爆発的な拡大により，大量のウェブページの中から有用な情報を抽出する技術はより重要さを増している．そのための手法として，近年**ウェブマイニング**（web mining）[1, 2] の研究が盛んに進められている．特に，ウェブページ間のリンク情報を用いる**リンク解析**（link analysis）とウェブページのテキストとタグ構造などを分析する**内容解析**（contents analysis）は重要な技術となっている．

一方では，ウェブページを効果的に収集するクローリング手法に関しては，これまでさまざまなアプローチが提案されており，特に，ユーザが着目しているトピックに関するウェブページを集中的に収集する**トピック主導型クローリング**（topic-focused crawling）については，数多くの研究がなされている [1, 2, 3, 4].

[†]筑波大学大学院
システム情報工学研究科コンピュータサイエンス専攻
Department of Computer Science,
Graduate School of Systems and Information Engineering,
University of Tsukuba
[††]筑波大学計算科学研究センター
Center for Computational Sciences,
University of Tsukuba

本稿では，ウェブマイニングの技術（特に内容解析およびリンク解析）とトピック主導型クローリングの技術を統合した，ユーザが提供する例示データに基づく選択的な情報収集手法を提案する．本手法の基本的アイデアは次のようになる．

1. ユーザが提供した例示データの集合およびシードのページ集合をもとに選択的なウェブのクローリングを行い，提供された例示データに関連する情報を保持するページを収集する．たとえば，あるユーザがデータベース関係の本の著者とタイトルの情報に興味を持っている場合，図1に示すようなデータを提供することが考えられる．以下ではこのような例示データを**レコード**と呼ぶ．
2. クローリング処理と並行して，取得されたページ中から新たなレコードを抽出し，ユーザが提供したレコードに追加する．また同時に，新たなレコード抽出パターンの学習も行う．
3. 新たに追加されたレコードについて，トピックの点で不適合なもの，および，データの抽出にミスがあるものについてフィードバックを受け

Author	Title
A.F. Cardenas	Data Base Management Systems
C.J. Date	An Introduction to Database Systems
D. Maier	The Theory of Relational Databases
H.S. Kunii	Graph Data Model and Its Data Language
R.G.G. Cattell	Object Data Management

図 1: 例示レコード集合

Fig. 1: Example record set

る．これをもとに，レコード抽出パターンの修正とクローリング処理の修正を図る．

このような処理により，ユーザとの対話に基づき，提供された例示データを拡充することが本手法の特徴となる．また，クローリングの際には，ウェブページ間のリンク情報のみではなく，データベース（例示レコードと抽出レコードを含む）とウェブページの関連性により，ウェブページのクローリングを制御する．この意味で，本アプローチはデータベース主導のクローリングとも位置付けられる．

2 関連研究

2.1 ウェブマイニング

ウェブマイニング [2, 1] は，大別すると，

1. ウェブページの内容解析
2. リンク解析
3. ログ解析

の 3 つのアプローチに分けることができる．本研究では，内容解析とリンク解析の融合を図る．

2.1.1 リンク解析

ウェブページ間のリンク情報を用いるリンク解析は，評判の高いウェブページを特定するための重要な技術となっている．リンク解析手法の代表例としては，Google で用いられている **PageRank** [5] や，ユーザが指定したトピックに関してハブとオーソリティのページを抽出する **HITS** (Hypertext Induced Topic Selection) [6] が挙げられる．

ウェブのリンク解析にヒントを得て，近年ではデータベースに対してもリンク解析を適用するアプローチが提案されている [7, 8]．データベース中に存在する関連情報を一種のリンクとみなしてリンク解析を行うもので，データベースの問合せ能力の活用などが課題となっている．本稿で提案する手法は，実際のデータベースを外部のウェブページと統合してリンク解析を行うという点で，これらのアプローチの拡張になっていると考えることができる．

2.1.2 内容解析（情報抽出）

DIPRE (Dual Iterative Pattern Relation Expansion) [9] は HTML 文書からレコード集合を抽出するためのアプローチである．例えば，いくつかの本の著者とタイトルのペアをユーザに与えられると，DIPRE は収集された HTML 文書集合の中から与えられたレコードを見つけるためのパターン集合を抽出し，それらのパターンを用いて新たなレコードの抽出を図る．ウェブ環境では，関連するレコードが HTML 文書に一定の文脈で繰り返して現れる傾向があるため，この手法は単純であるがうまく作動するといわれる．後述のように，本研究では，情報抽出に DIPRE を利用する．

他の手法の例を挙げる．**Snowball**[10] はプレーンテキスト文書からパターンを生成し，リレーションを抽出する．DIPRE より厳密なパターンとリレーションの評価基準を提案している．パターンは文字列のみではなく，固有表現（組織名，地名など）も利用している．**MDR**[11] はユーザのサポートは必要がなく，多数のデータレコードを含むたった一つのページから，自動的にレコードを抽出する．この手法は木構造に基づくアルゴリズムを利用している．

2.2 クローリング

ウェブのクローリング方式においては，従来よりさまざまな方式が工夫されてきた．効率的なクローリングには重要なページを優先的に辿ることが求められる．クローリングの効率化に加え，近年，**トピック主導型クローリング**（topic-focused crawling）が着目されている [3, 4, 12, 13, 14]．これは，与えられたトピックに対するページを効率よく収集することを目的としている．最良優先（best-first）クローラ [12] では，アクセスしたページとクエリとの類似度を計算し，類似度の高いページ内の URL を優先的にアクセスする．PageRank クローラ [12] は，PageRank [5] によるスコアに基づいてページのアクセスの順序を決めるアプローチである．[13, 14] では，分類器を用いて選択的にクローリングを行う手法が提案されて

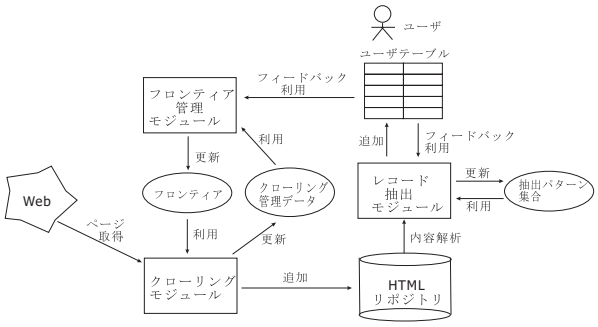

図 2: システムの構成

Fig. 2: System architecture

おり，[3] では複数のトピック主導型クローリングにおける能力の比較を行っている．

本研究の目的は，提供されたレコードに関連するウェブページを優先的に効率よく探索し，データベースとウェブの融合を図る点にあり，トピック主導型クローリングと関連が深い．

3 システムの概要

図 2 をもとに，想定するシステムの構成について述べる．

ユーザが最初に与えた例示レコードは，**ユーザテーブル**（user table）に収められる．システムによるその後のレコードの追加も，このテーブルが対象となる．

選択的なクローリングにおいて，クローリングの方針を決め，どのページを優先して辿るかを決定するのが**フロンティア管理モジュール**（frontier management module）である．このモジュールは，取得された HTML ページの集合とユーザテーブル内のレコードの情報をもとに，ユーザが求めるレコードを含むページを優先的に探索するための**フロンティア**（frontier）の管理を行う．フロンティアは順序付けされた URL のリストであり，フロンティア管理モジュールにより逐次更新される．なお，初期時点では，ユーザから与えられたシードの URL の集合をもとにフロンティアが作られるものとする．

実際にクローリングを行うのが**クローリングモジュール**（crawling module）である．このモジュールは，フロンティアの内容をもとに，ウェブページをダウンロードし，**HTML リポジトリ**（HTML repository）に追加する．また，リンク抽出などを行い，**クローリング管理データ**を更新する．クローリング管理データはフロンティア管理モジュールと共有され，フロンティアの維持管理に利用される．

レコード抽出モジュール（record extraction module）は，HTML リポジトリに追加されたページからのレコード抽出を行う．初期時点では，ユーザから与えられた例示レコードを抽出するためのパターンを学習し，その結果をもとに HTML リポジトリから新たなレコードの抽出を図る．新たなレコードはユーザテーブルに適宜追加される．ユーザは，追加されたレコードに対し適宜フィードバックを行う．ユーザテーブルをブラウジングし，着目したレコードに対し，以下のどれに相当するかをシステムに通知する．

1. 追加されたレコードが適合である
2. 追加されたレコードのトピックが不適合である：たとえば，データベース分野の本の著者とタイトル情報が求められているのに，ビジネス分野の本の著者とタイトルが含まれている場合などである．
3. 追加されたレコードにノイズが含まれている：レコードの抽出パターン自体に問題があり，レコードとして不適切なデータが抽出された場合にあたる．

フィードバックは，(2) の場合にはフロンティア管理モジュールに送られ，(3) の場合にはレコード抽出モジュールに送られる．それぞれのモジュールがユーザからのフィードバックに応じて，フロンティア算出手法の修正や抽出パターンの見直しを行う．

なお，フロンティア管理モジュール，クローリングモジュール，レコード抽出モジュールは独立して並行に動作する．これにより，クローリング処理と同時にレコードの抽出やフロンティアの見直しなどが行われることになる．

4 レコード抽出手法

本研究では，情報抽出には **DIPRE** (Dual Iterative Pattern Relation Expansion) [9] の利用を想定している．DIPRE の処理ステップは以下のようになる．

1. シードとなるレコード集合が与えられる．
2. HTML リポジトリから，シード集合に対応するレコードの**オカレンス**（occurrence）を見つける．オカレンスは

   $(url, attributes, order, prefix, separators, suffix)$

という形式で表される．url はレコードが見つかった URL を表し，$attrbutes$ は抽出対象のレコードの配列を表す．たとえば図 1 の例の場合，$attributes$[0] が著者名（例：C. J. Date）に，$attrbutes$[1] がタイトル（例：An Introduction to Database System）に相当することになる．$order$ は，その HTML ページ内で属性がどの順序で出現したかを表す属性である．$prefix$ と $suffix$ はそれぞれ，最初および最後に出現する属性の前および後に出てくるタグ等のパターンである．$separators$ は属性間の区切りのパターンに相当し，図 1 の場合は $separators$ が著者とタイトル（あるいはタイトルと著者）を区切るパターンを保持することになる．

3. 発見されたオカレンスの集合をもとにパターン集合を生成する．パターンは

   ($order$, $urlprefix$, $prefix$, $separators$, $suffix$)

   の形式を持つ．パターン生成においては，まず，オカレンス集合を同じ $order$ と $separators$ を持つオカレンスごとにグループ化する．含まれるオカレンスの数が 1 件しかないグループは削除し，残りの各グループについてパターン生成を試みる．

   パターン生成においては，グループ内のすべてのオカレンスについて，url の最長接頭辞，$prefix$ の最長接尾辞，$suffix$ の最長接頭辞を抽出し，それぞれをパターンの $urlprefix$, $prefix$, $suffix$ とする．これらのいずれかが空になる場合，グループを合併して再度パターン抽出を試みる [5]．

4. 追加されたパターン集合が得られると，これをもとに HTML リポジトリ中から再びレコードのオカレンスを抽出する．その結果をもとにパターン集合を更新する．このような処理を繰り返すことで，逐次的にレコードを抽出する．

DIPRE では，静的な HTML リポジトリを対象とし，収束するまでレコードの抽出を繰り返すということを基本としていた．しかし，本研究においては，HTML リポジトリにページが新たに追加されることを想定している．そのため，レコード抽出モジュールは，新たなページからレコードのオカレンスの抽出を試みるか，新しいパターンで既存の HTML ページを探索しなおすかを，適宜判断することが必要となる．このための戦略の構築は今後の課題の一つである．

また，レコード抽出モジュールは，ユーザからのフィードバックにも対応する必要がある．抽出結果に問題があると指摘された場合，レコード抽出モジュールは，該当するレコードの抽出に用いられたパターンを，今後のレコード抽出に用いないように抹消する処理を行う．

5 クローリング手法

5.1 基本的な考え方

クローリングモジュールは，単純にフロンティアに登録された URL のリストの上位から順にクローリング処理を行う．よって，フロンティアを維持管理するフロンティア管理モジュールの戦略がクローリング手法を規定することになる．本研究では，ユーザテーブルに蓄積されたレコードの集合をもとに，選択的なクローリングを行うことを目的としている．そこで，レコード集合をどのようにフロンティア管理モジュールの戦略に反映させるかが重要なポイントとなる．

以下では，ユーザテーブル中のレコード集合を考慮してクローリングを行うための 3 つのアプローチを提案する．特に第 3 のアプローチに重点を置いて説明する．その前に，次節ではこれらのアプローチに独立な汎用のクローリング処理について述べる．

5.2 クローリング処理

以下で共通の，汎用のクローリング処理をアルゴリズム 1 に示す．未訪問の URL の集合であるフロンティアをどのようにランク付けするかの戦略により，クローリングモジュールの動作が決定する．なお，ここには明記していないが，クローリングモジュールとは独立に，レコード抽出モジュールとフロンティア管理モジュールが並行に動作している．よって，フロンティアのランク付けは動的に変化しうる．

5.3 テキスト化によるアプローチ

レコード集合全体をテキストとみなし，これをユーザが与えたキーワード集合であるとみなせば，既存のトピック主導型クローリングの手法の多くが利用できる．たとえば，**最良優先**（best-first）法は，一般に，ウェブページを単語のベクトル $\boldsymbol{d}$ で表現する [3, 12]．ユーザの問合せ $\boldsymbol{q}$ との類似度 $sim(\boldsymbol{q}, \boldsymbol{d})$ を計算し，

Algorithm 1 Crawling Algorithm

```
1: アクセスした URL の集合を crawled_urls とする.
2: 未アクセスの URL の集合を frontiers とする.
3: while not empty(frontiers) do
4:    frontiers 中でスコアが最大の URL を url とする.
5:    url を frontiers から削除する.
6:    page := get_page(url)
7:    append(crawled_urls, url)
8:    url_list := extract_urls(page)
9:    for all u ∈ url_list do
10:      if u ∉ crawled_urls and u ∉ frontiers then
11:         frontiers := frontiers ∪ {u}
12:      end if
13:   end for
14: end while
```

上位にランクされたページについて，未訪問の URL を優先的に探索する．このアプローチを本手法に関して拡張すると，レコード集合全体をテキストデータとみなし，問合せベクトル $\boldsymbol{q}$ を構成することになる．新たなレコードの追加が生じると，$\boldsymbol{q}$ も更新されることになる．

SharkSearch [4] や，分類器を用いるアプローチ [13, 14] などもこのような考え方を用いれば適用可能になると考えられる．ただし，ユーザテーブル中のレコードには，たとえば “C. J. Date” などのように，単語として分解してしまうと意味をなさないものもある．この点で，レコードに着目する本研究のアプローチを直接的に活かせない可能性がある．

ユーザからのフィードバックにより，ユーザテーブル中のレコードが不適合と指示された場合には，情報検索でしばしば用いられる**適合性フィードバック**（relevance feedback）[15] のアプローチを用いて，問合せベクトル $\boldsymbol{q}$ を修正することが考えられる．

5.4 レコードの共有率を考慮したアプローチ

上記のアプローチでは，どのページが実際にレコードを多く含んでいるかという点に対する配慮がなされていなかった．HTML リポジトリ中の各ページからどのレコードが抽出されたを記録しておくことで，以下に述べるようなフロンティアのランク付けが可能となる．

まず，各ページ p について，そこから抽出されたレコードの集合を $records(p)$ で表す．このとき，ユーザテーブル t に関するページ p のスコアを，Jaccard 係数に基づき

$$score(t,\, p) = \frac{|t \cap records(p)|}{|t \cup records(p)|} \qquad (1)$$

で与えることができる．これにより，ページ中にユーザテーブル中のレコードが多く含まれるほど，ページ p のランクは高くなることになる．また，ユーザにより不適合とフィードバックを受けたレコードをページ p が多く保持している場合，p のスコアが低くなるという効果も得られる．

このアプローチの問題点は，レコードを含まない多くのページについてスコアが 0 となると考えられることである．そこで，このアプローチにおいては，スコアが高いページ群をシードページの集合とみなすことで，既存のトピック指向型クローリング手法の適用を行う．シードページの集合からテキストのベクトル $\boldsymbol{q}$ を作成すれば，上述の最良優先法が利用できる．また，PageRank によるクローリングのアプローチ [12] なども適用できることになる．

5.5 ObjectRank に基づくクローリング手法

5.5.1 基本的なアイデア

近年，ウェブのリンク解析にヒントを得て，データベースに対してもリンク解析を適用するアプローチが提案されている [7, 8]．ウェブを対象とするのではなく，データベース中のデータに対し，データの関連性をもとにデータの重要性を算出することがその狙いである．ここでは，その一つである ObjectRank [7] を拡張するアプローチを提案する．

その概要は次のようになる．

1. まず，**レコード－ページグラフ**（record-page graph, RPG）を構築する．RPG のノードは，ユーザテーブルの各レコードと各ウェブページからなる．また，RPG の辺は，ウェブページ間のハイパーリンクと，レコードとウェブページ間のリンクからなる．RPG は，クローリングに伴うページ収集とレコード抽出に応じて拡張される．

2. RPG において，各ノードのスコアを計算する．このスコアを [7] に従い，**ObjectRank** と呼ぶ．各ページに対し求められた ObjectRank のスコアにより，次に探索するページの選択を行う．ObjectRank スコアの計算にはユーザテーブル中のレコード集合との関連を考慮しているため，クローリング処理がユーザが関心を持つページを中心に行われると期待される．ObjectRank のスコアも，ページの取得とレコードの追加に応じて再計算される．

トピック主導型のクローリングにおいては，シードとして与えられたページ群をもとに，リンクによる参照とページの内容に基づいてページが選択的に収集される．一方，本手法におけるクローリングは，ユーザテーブル中のレコード集合をもとに行う点に特色がある．ユーザテーブル中のレコードはユーザの興味を表している．したがって，本アプローチにより，ユーザの関心を持つウェブページを収集できると考えられる．

以下では，提案手法の詳細について述べる．

5.5.2 RPG の構築

ウェブのクローリングを開始する前に，ユーザテーブル中のレコードと，それに関連するウェブページの関係を表現する RPG を以下の手順で構築する．

1. ユーザからシードとして与えられたページ集合の各ページについて，4 節で述べたアプローチによりレコード抽出を行う．新たなレコードが抽出された場合は，ユーザテーブルに追加する．
2. ユーザテーブルの各レコードに対しノードを生成する．
3. シードのページ集合の各ページについてノードを生成する．
4. シード集合の各ページについて，リンクの抽出を行い，未アクセスの URL に対してノードを生成する．
5. ステップ 4 でのリンク抽出の際に，ページ間の参照関係に基づいて，グラフの辺を生成する．ページ p がページ q へのハイパーリンクを有するとき，$p \to q$ という辺を生成する．

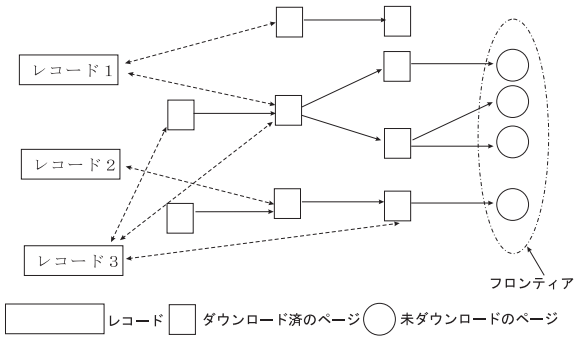

図 3: RPG の例

Fig. 3: RPG (record-page graph)

6. ステップ 1 において，ページ p からレコード集合 $records(p)$ が抽出されたとき，各レコード $r \in records(p)$ について $r \to p$ および $p \to r$ という 2 つの辺を生成する．

[7] のアプローチにしたがい，RPG の辺には**重み**（weight）を設定する．辺の重みの値はリンク解析に反映される．RPG には，ステップ 5 で生成されるページ間の辺，ステップ 6 で生成されるレコードからページへの辺，および，ページからレコードへの辺の 3 種類の辺が存在する．それぞれについて，w_{pp}, w_{rp}, w_{pr} の値を与える．これらはユーザにより指定される定数で，1 以下の正の値とする．なお，同一ノードから同じ種類の複数の辺が出る場合には，重みを各辺で等分する．たとえば，あるページ p に対応するノードから 3 つのページ間の辺が出る場合，各辺には $w_{pp}/3$ の重みを割り当てる．

以上のようなアプローチにより，図 3 のような RPG のグラフ構造を得る．なお，ユーザから最初に与えられたシードページの集合中に，ユーザが指定したレコードが一切含まれていないという場合も発生しうる．そのような場合には，初期時点ではレコードに対応するノード群とページに対応するページ群の間に辺が存在しないような RPG が構築される．ただし，ユーザが与えたシードページ集合がトピックの面で適切であれば，ある程度のクローリング処理の後には，該当するレコードを含むウェブページが取得でき，図 3 のような RPG が構築されると考えられる．

5.5.3 ObjectRank 値の計算

前節で構築された RPG において，各ノードの ObjectRank 値を計算する．ObjectRank は，RPG 上を

遷移する random surfer の各ノードへの訪問確率に相当し，RPG から見たノードの重要度を表す．その値は以下の式で計算する．

$$\boldsymbol{r} = d\mathbf{A}'\boldsymbol{r} + \frac{1-d}{n}\boldsymbol{e} \tag{2}$$

n は RPG のノード数である．$\mathbf{A}$ は，$n \times n$ 行列であり，その (i, j) 要素はノード i からノード j への辺の重みの総和である．$\mathbf{A}'$ は，$\mathbf{A}$ の転置行列を表す．$\boldsymbol{r}$ は n 次元列ベクトルであり，各ノードの ObjectRank 値を表す．$\boldsymbol{e}$ はすべての要素が 1 である n 次元列ベクトルである．d はユーザから指定される定数（*damping factor*）であり，辺に沿ったノード間の移動とランダムな移動を調整する．ObjectRank 値は，$\boldsymbol{r}$ の各要素を 1 に初期化し，上の式を収束するまで評価することで得ることができる．ObjectRank の計算式は PageRank と同様だが，計算は通常のウェブグラフではなく，RPG において行うという点が異なる．

5.5.4 フィードバックへの対応

ユーザから，ユーザテーブル中のレコードに対し不適合というフィードバックが与えられたときの対応策について述べる．不適合と指定されたレコードに対し，そのようなレコードを含んでいるページがスコアの面で損をするような設定にすることで，ユーザのフィードバックを反映することが基本的なアイデアである．

図 4 を用いてアイデアを説明する．左側の図は RPG のサブグラフを想定しており，レコード r_1, r_2 が，それぞれページ p_1, p_2, p_3 を参照している．レコードのランクを $rank(r_i)$ $(i = 1, 2)$，ページのランクを $rank(p_j)$ $(j = 1, 2, 3)$ と表す．式 (2) の 1 回の計算において，直感的には各ページのスコアが

$$\begin{aligned}
rank(p_1) &= d\frac{w_{rp}}{2} \cdot rank(r_2) + \frac{1-d}{n} \\
rank(p_2) &= d\frac{w_{rp}}{2} \cdot (rank(r_1) + rank(r_2)) + \frac{1-d}{n} \\
rank(p_3) &= d\frac{w_{rp}}{2} \cdot rank(r_1) + \frac{1-d}{n}
\end{aligned}$$

と計算され，レコードのスコアが伝播することになる．

ここで，レコード r_1 がユーザにより不適合と指定されたとする．このとき，単純なアプローチとしては，r_1 に対するノードを削除し，RPG を再構築することが考えられる．これは，間接的にはレコード r_1 を含む p_2, p_3 の評価を下げることにつながるが，レコード r_1 を指すことがペナルティとはならない．

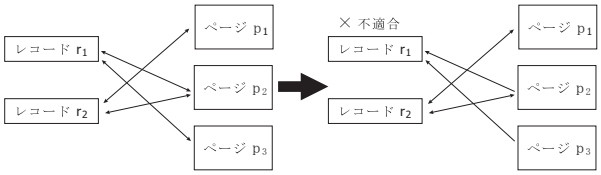

図 4: フィードバックへの対応

Fig. 4: Processing of user feedback

そこで，図 4 右のように，レコード r_1 のノード自体は削除せず，r_1 から各ページに出る辺のみを削除する．この例では，$r_1 \to p_2, r_1 \to p_3$ という 2 つの辺を削除する．これにより，スコアは

$$\begin{aligned}
rank(p_1) &= d\frac{w_{rp}}{2} \cdot rank(r_2) + \frac{1-d}{n} \\
rank(p_2) &= d\frac{w_{rp}}{2} \cdot rank(r_2) + \frac{1-d}{n} \\
rank(p_3) &= \frac{1-d}{n}
\end{aligned}$$

と計算することになり，r_1 からのスコアの伝播がなくなる．一方，ページ p_2 のスコアはレコード r_1, r_2 に伝播し，ページ p_3 のスコアはレコード r_1 に伝播する．よって，ページ p_2, p_3 のスコアには損失が生じ，特に p_3 については，ObjectRank 計算の繰り返しにより，最終的なスコアは $rank(p_3) = (1-d)/n$ に収束する．

以上のアプローチにより，ユーザからのフィードバックを反映して ObjectRank を再計算することが可能となる．

6 まとめと今後の課題

本稿では，ユーザが与えた例示レコードをもとに，関連するレコードを拡充するための適応型のクローリング手法の枠組みについて述べた．システムの構成方式を示し，レコード集合の情報をクローリング処理に反映するための 3 種類のクローリング手法を提案した．また，発見された新たなレコードにユーザがフィードバックを行い，ユーザの観点からみて不適合であるレコード抽出にユーザの意見を反映させたり，レコード抽出パターンに問題がある場合のパターンの修正などを行うアプローチを示した．

今後の課題としては以下が挙げられる．

1. システムの実装と実験：提案したシステム構成をもとに実装を行い実験を行う．そのためには，

クローリング手法およびレコード抽出手法の詳細化が必要となる．

2. レコード抽出法に関しては，現時点では DIPRE [9] の利用を想定しているが，MDR [11] などのより精度のよい手法も提案されている．しかし，本手法はクローリングとレコード抽出を組み合わせる点に特徴があり，処理時間の大きいレコード抽出法は不向きであると考えられる．このような面を考慮して，レコード抽出法の改善を図る．

3. 本稿では議論の簡単化のため，単純なクローラの構造をもとに，レコード集合に基づくフロンティアのランク付け手法を導入するアプローチを提案した．しかし，高度化し大規模化したサーチエンジンを効率的に利用することも考えられる．サーチエンジンをシステム構成の一部に取り込んだ際の，効果的な実現手法についても検討を行う必要がある．

謝辞

本研究の一部は，日本学術振興会科学研究費基盤研究 (C)(16500048)，同 (B)(15300027)，旭硝子財団研究助成，稲森財団研究助成，文部科学省科学研究費特定領域研究 (2)(16016205) 及び CREST「自律連合型基盤システムの構築」による．

参考文献

[1] P. Baldi, P. Frasconi, and P. Smyth, *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*, Wiley, 2003.

[2] S. Chakrabarti, *Mining the Web: Analysis of Hypertext and Semi Structured Data*, Morgan Kaufmann, 2002.

[3] F. Menczer, G. Pant, and P. Srinivasan, Topical Web Crawlers: Evaluating Adaptive Algorithms. *ACM Trans. on Internet Technology*, Vol. 4, No. 4, pp. 378–419, 2004.

[4] G. Pant, P. Srinivasan, and F. Menczer, Crawling the Web, *Web Dynamics: Adapting to Change in Content, Size Topology and Use*, M. Levene and A. Poulovassilis (eds.), Springer-Verlag, pp. 153–177, 2004.

[5] S. Brin and L. Page, The Anatomy of a Large-scale Hypertexual Web Search Engine. *Computer Networks*, Vol. 30, No. 1-7, pp. 107–117, 1998.

[6] J.M. Kleinberg, Authoritative Sources in a Hyperlinked Environment. *JACM*, Vol. 46, No. 5, pp. 604–632, 1999.

[7] A. Balmin, V. Hristidis, and Y. Papakonstantinou, ObjectRank: Authority-Based Keyword Search in Databases. *Proc. VLDB*, pp. 564–575, 2004.

[8] F. Geerts, H. Mannila, and E. Terzi, Relational Link-Based Ranking. *Proc. VLDB*, pp. 552–563, 2004.

[9] S. Brin, Extracting Patterns and Relations from the World Wide Web. *Proc. WebDB*, 1998.

[10] E. Agichtein and L. Gravano, Snowball: Extracting Relations from Large Plain-Text Collections. *ACM DL*, 2000.

[11] B. Liu, R. Grossman, and Y. Zhai, Mining Data Records in Web Pages. *Proc. KDD*, 2003.

[12] J. Cho, H. Garcia-Molina, and L. Page, Efficient Crawling through URL Ordering. *Computer Networks*, Vol. 30, No. 1–7, pp. 161–172, 1998.

[13] S. Chakrabarti, M. van den Berg, and B. Dom, Focused Crawling: A New Approach to Topic-specific Web Resource Discovery. *Computer Networks*, Vol. 31, No. 11–16, pp. 1623–1640, 1999.

[14] M. Diligenti, F. M. Coetzee, S. Lawrence, C.L. Giles, and M. Gori, Focused Crawling Using Context Graphs. *Proc. VLDB*, pp. 527–534, 2000.

[15] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.