Vol. 48 No. SIG 14(TOD 35) 情報処理学会論文誌：データベース Sep. 2007

トピックを考慮した大規模文書情報源からのレコード抽出

張 建偉[†] 石川佳治[††] 北川博之[†,†††]

近年，大量のテキスト文書からのレコード抽出の研究が行われている．レコード抽出には次の課題が存在する．第1に，大量の文書を情報抽出の対象とした場合に多大な処理コストがかかる．第2に，抽出されたレコードが，必ずしもユーザが興味あるトピックと合致しないことがある．これに対し本稿では，ユーザの意図に合った情報を効率良く抽出するためのレコード抽出手法を提案する．本手法では，効果的な抽出のために，ユーザの意図に適合した情報を含んでいる可能性の高い文書群を特定する．その特定した文書群を優先的に抽出処理に利用することで処理コストの削減を目指す．また，それらの文書群から内容の関連が深いレコードを抽出することで高い抽出精度を達成する．実験結果により，提案手法が抽出精度の低下を防ぎつつ，処理コストの削減を実現できることを示す．

Record Extraction from Large-scale Text Resources Considering Topics

Jianwei Zhang,[†] Yoshiharu Ishikawa[††]
and Hiroyuki Kitagawa[†,†††]

In recent years, the research on record extraction from a large number of text documents is becoming popular. However, there still exist some problems in record extraction. 1) When a large number of documents are used for the target of information extraction, the process usually becomes very time-consuming. 2) It is also likely that extracted records may not pertain to the user's interest on the aspect of the topic. To address these problems, in this paper we propose a method for efficiently extracting those records whose topics are relevant to the user's interest. To improve the efficiency of the information extraction system, our method identifies documents from which useful records are probably extracted. Those selected documents are first processed in order to reduce processing cost. Moreover, from these documents user-desired records are apt to be extracted so that high extraction accuracy is obtained. Our experiments show that our system reduces the processing cost with achieving high extraction accuracy.

1. はじめに

近年，様々な情報発信手段の発達により，電子化されたテキスト文書が急激に増加している．このようなテキスト文書には有用な情報がしばしば含まれている．しかし，これらの情報は構造化されておらず，計算機では容易に扱うことができない．そのため，テキスト文書から有用な情報を自動的に，あるいは半自動的に抽出する**情報抽出**の研究が近年重要視されている．そのうち，**ブートストラッピング型**と呼ばれる情報抽出手法が注目を浴びている[1)~4)]．これは抽出パターン（テンプレート，ルールとも呼ぶ）と抽出レコード（タプル，語とも呼ぶ）を交互に繰り返し学習することにより，少数のサンプルレコードから大量のレコードを抽出する手法である．直感的には，**レコード**とは1つないし複数の属性からなるデータで，各属性には同種のデータが含まれるようなものをいう．本稿では，このような構造化されたレコード構造の情報を抽出する試みに着目する．抽出されたレコードの集合は一種のデータベースと考えられ，既存のデータベースとの統合など，様々な形で応用することが可能となる．

従来のブートストラッピング型の情報抽出の研究は，大量のレコードの獲得やレコードのノイズの削減といった問題に着目してきた．しかし，それ以外にも，以下のような2つの問題がある．1つ目の問題は，一般的に情報抽出システムでは，文書に対しタグ付けなどの前処理を行ったり，文書をスキャンしたりする必要があるため，テキスト文書の量が非常に多い場合，

† 筑波大学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
†† 名古屋大学情報連携基盤センター
Information Technology Center, Nagoya University
††† 筑波大学計算科学研究センター
Center for Computational Sciences, University of Tsukuba

多大な処理コストがかかるということである．2つ目の問題は，単なるパターンマッチングで抽出されたレコードは，必ずしもユーザが興味あるトピックと合致しないことである．たとえば，IT関係の会社と場所のペアを抽出しようとする場合，他のトピックのレコード（たとえば，自動車関係の会社と場所のペア）が抽出されても，ユーザは満足できないと思われる．

本研究ではこれに対し，ユーザの意図に合った情報を効率良く抽出するためのレコード抽出手法を提案しており，少ない処理コストでユーザがほしいレコードを早く獲得することを目指す．1つ目の問題の解決に対しては，一般的に大量の文書の中で一部の文書のみが情報抽出タスクに関連していることに着目する．本手法では効率的な抽出のために，ユーザの意図に適合した情報を含んでいる可能性の高い文書群を特定し，選択された文書を優先的にレコード抽出の対象とする．大量の文書全体を区別せずに処理するのではなく，求められる情報の抽出に有用な文書を優先的に処理することにより，レコード抽出の効率を向上させる．2つ目の問題について，本研究では，レコードにノイズが存在するかどうかだけではなく，ユーザの意図に合致するかどうかも考慮する．提案手法では，着目するトピックに適合する文書を選択するため，そこから抽出されるレコードのトピックも正しい可能性が高くなり，より高い抽出精度が達成できる．

本稿では，まず，2章で関連研究について述べ，3章において既存のレコード抽出手法を説明する．4章では，提案手法について説明する．次に，5章では，提案手法の効果を実験によって評価し，本手法の有用性を検証する．最後に，6章においてまとめと今後の展開を述べる．

2. 関連研究

ウェブページやニュース記事などのテキスト文書からの情報抽出の研究には，様々なアプローチがある．

(1) 構造に基づくウェブ情報抽出手法

Cravenらが提案した情報抽出システム[5]は，オントロジと学習データが与えられると，機械学習の手法を利用して知識ベースを構築する．Lixto[6]はユーザが抽出パターンを指定できる視覚的なウェブ情報抽出システムである．Kushmerickが提案した手法[7]は，人手によりラベル付けされたウェブページの集合から，機械学習の手法を利用して抽出パターンを学習する．これらの研究は1つのページから，あるいは構造が類似するページから情報を抽出するアプローチである．

(2) ブートストラッピング型の情報抽出手法

ブートストラッピング型の手法[1)~4)]は，例示情報をもとに，大量の文書から自動的に情報を抽出することを目指す．ブートストラッピング型のレコード抽出法の1つに**DIPRE**（Dual Iterative Pattern Relation Extraction）[1]がある．もともとはウェブページ群からレコード集合を抽出するための手法として提案された比較的単純な手法である．基本的な考え方は，抽出パターンと抽出レコードを交互に学習することである．抽出パターンの生成では，レコードの出現コンテキストだけではなく，レコードが発見されたURLのパターンも分析する点に特徴がある．ウェブ環境では，関連するレコードがウェブページ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機能するといわれる．Zhangらの研究[2]では，DIPREの繰返し処理回数を減らすためのレコード抽出手法を提案している．毎回のレコード抽出処理を実行した後，文書全体から抽出可能なレコード数を予測する．そのうちすでに抽出されたレコードの割合を計算し，与えられた閾値に達成したら繰返し処理を終了する．**Snowball**[3]はDIPREを拡張したアプローチである．DIPREとは異なり，主として構造化されていないプレーンテキスト文書からのレコード抽出を対象としている．特徴の1つは，固有表現を利用することである．固有表現を用いた抽出パターンを生成することにより，精度の高い抽出処理を実現する．楠村らの研究[4]では，DOM構造の解析とパターンの交叉によるウェブ上の表や箇条書からもレコードを抽出できる手法を提案している．

(3) トピック主導型のクローリング

一方，クローリングの効率化に関する研究として，トピック主導型のクローリングが着目されている[8),9)]．これらの研究は，あらかじめ欲しいページのトピックを与えて，そのトピックに合致するページを重点的に収集することを目的としている．関連するページを取得し，無関係なページを無視するという点がトピック主導型のクローリングのポイントである．

本研究では，(3)のアイデアにヒントを得て，抽出に役立つ文書を選び，それらの文書のみに対して抽出のための分析を行う．レコード抽出処理に抽出対象文書を選択するプロセスを導入することで，(2)で述べたDIPREとSnowball手法の基本的なフレームワークを拡張する．処理コストの削減に着目したメタなレコード抽出手法としてQXtract[10]が提案されているが，抽出レコードがトピックに関してユーザの興味を満たすかどうかを考慮し，文書を選択する点において，

本研究は彼らの研究と異なる．

3. ブートストラッピング型のレコード抽出手法

この章では，まず一般的なブートストラッピング型のレコード抽出手法について述べる．具体的にはDIPRE法を拡張した**Snowball**の手法がベースになっている．**図1**で示されるように，抽出したいレコードのサンプルとして例示レコード集合がユーザによって提供される．この手法は，例示レコード集合に基づいて文書リポジトリ内の文書を分析し，新たなレコードの抽出を図る．繰返し処理により，少数のサンプルレコードから大量のレコードを抽出することがブートストラッピング型の手法の特徴である．処理の概要を**アルゴリズム1**に示す．

Algorithm 1 ブートストラッピング型手法に基づくレコード抽出

1: *Seed*：例示レコード集合
2: *Doc*：文書リポジトリ
3: $Doc\_tag = attach\_tag(Doc)$
　{ 固有表現タグ付け }
4: **repeat**
5: 　$Occ = find\_occurrences(Doc\_tag, Seed)$
　{ オカレンスを発見 }
6: 　$Pat = generate\_patterns(Occ)$
　{ パターンを生成 }
7: 　$Rec = extract\_records(Doc\_tag, Pat)$
　{ レコードを抽出 }
8: 　$Sorted\_Rec = sort\_records(Rec)$
　{ レコードをランク付け }
9: 　$Seed = Seed \cup Top\_k(Sorted\_Rec)$
　{ 上位 k 件のレコードを追加 }
10: **until** *termination criterion*
11: **return** *Rec*

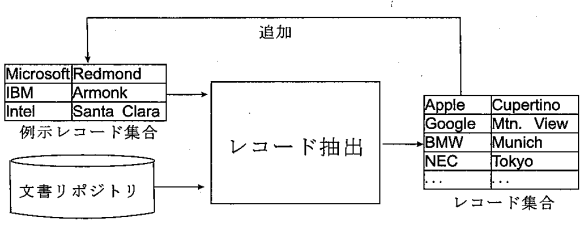


図1 レコード抽出
Fig. 1 Record extraction.

処理の過程は次のようになる．

(1) **例示レコード集合の提示および固有表現タグ付け**（1〜3行目）：ユーザから例示レコード集合が与えられる．これはユーザが獲得しようとする目標レコードを反映している．図1の例では，ユーザは会社と所在地のペアに興味を持っていると想定している．前処理として，固有表現抽出器[11]を用いて文書中に出現した人名，組織名，地名などを認識する．

(2) **オカレンスの発見**（5行目）：文書リポジトリから，例示レコード集合に対応するレコードのオカレンスの集合を見つける．オカレンスはレコードの属性が文書内でマッチした際のコンテキストであり，下記のような形式で表される：
$(company, location, o\_prefix, tag1, o\_middle, tag2, o\_suffix)$
company は抽出対象のレコードの会社の名前であり，*location* は会社の場所である．*o_prefix* と *o_suffix* は与えられたレコードの属性がテキストにマッチした際の，それぞれ前方および後方のテキストのパターンである．*o_middle* は属性間の区切りのパターンに相当する．*tag1* と *tag2* は固有表現タグであり，この例の場合，*tag1* と *tag2* は組織名（ORG）や地名（LOC）の値をとる．たとえば，図1の例にある (*Microsoft*, *Redmond*) レコードがタグ付けされた文章 “Many researchers at ⟨ORG Microsoft⟩ in ⟨LOC Redmond⟩ have worked ...” に出現した場合，オカレンスは
$(Microsoft, Redmond, \text{“}at\text{”}, ORG, \text{“}in\text{”}, LOC, \text{“}have\text{”})$
というタプルに相当する．

(3) **パターンの生成**（6行目）：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
$(p\_prefix, tag1, p\_middle, tag2, p\_suffix)$
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 *tag1*，*o_middle* と *tag2* 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閾値以下であるグループは削除し，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，*o_prefix* の最長接尾辞，*o_suffix* の最長接頭辞を抽出し，それぞれをパターンの *p_prefix* と *p_suffix* とする．オカレ

ンスの *tag*1, *o_middle* と *tag*2 をパターンの *tag*1, *p_middle* と *tag*2 とし，パターンを生成する☆.

(4) **レコードの抽出**（7行目）：パターンが生成されると，パターンを用いて文書リポジトリを再度スキャンし，新たなレコードの抽出を試みる．

(5) **レコードのランク付け**（8行目）：自動的に抽出されたレコードにはノイズが含まれるものがあるという問題が存在する．ノイズがあるレコードが例示レコード集合に追加されると，それ以降に間違ったパターンが生成されやすいと考えられる．そのため，抽出されたレコードをランク付けし，ノイズがなく質の高いレコードを優先的に利用することが必要となる．

Snowball ではパターンとレコードの評価手法を提案しているが，抽出レコード集合において関数従属性が成り立つことを暗黙に仮定している．たとえば，(*Microsoft*, *Redmond*) と (*Microsoft*, *New York*) が抽出されると，そのレコードを抽出したパターンの信頼度を下げる．しかし，このような仮定は現実的に必ずしも成り立たない．そこで，本稿では後述の予備実験（5.2 節）により，複数のレコードランク付け手法の比較を行い，より効果的な手法を利用する．

(6) **例示レコード集合の拡充**（9 行目）：上位にランク付けされたレコードを例示レコード集合に追加し，次の繰返しを開始する．

終了条件（10 行目）が満たされるまで，以上のような処理を繰り返し行うことで，少ないサンプルレコードから大量のレコードが獲得できる．終了条件としては，文書リポジトリにおいて抽出されたレコード数が収束することなどがあげられる．

4. 提案手法

4.1 既存抽出手法の問題点と提案手法の目的

前章で述べたレコード抽出手法においては，固有表現を利用することで抽出の精度を上げることができるが，固有表現抽出のため処理のコストが大きくなる．後述の実験で利用した固有表現抽出器の場合，5,000 件の文書に固有表現タグを付けるのに 4.65 時間がかかる．1 つの文書の固有表現タグ付けに平均 3.3 秒ほどが必要である．そのため，実験に利用した 173,039 件のニュース記事の全文書に対して，固有表現タグ付けの処理時間が 6 日間以上にもなることが予測される．また，オカレンスの発見とレコードの抽出のためには文書をスキャンするのにも処理時間がかかる．すなわち，文書リポジトリが大きい場合，固有表現タグ付けや文書分析に多大な処理コストを要する．

一方，一般的にユーザはある特定のトピックに関する情報に興味を持ち，そのトピックに関連したレコードのみを抽出したい場合が多いと思われる．多くのレコードが抽出されるものの，そのうちユーザの興味に合致するレコード数が少ない場合，ユーザの立場からは抽出精度が良いとはいえない．文書リポジトリに複数のトピックの文書が含まれる場合，単なるパターン照合処理では，着目するトピック以外のレコードが抽出されることが避けられない．文書リポジトリの全文書を分析し，すべてのレコードを抽出した後，分類などの手法で関連しないレコードを事後に排除することが考えられるが，効率的な方法ではない．

本稿では，できるだけ少ない処理コストでユーザの興味に合致するレコードを抽出する手法の提案を目的とする．提案手法では，適切なレコード抽出用文書を求めることに焦点を当てている．ユーザの欲しい情報が含まれる可能性の高い文書を選択し，それらの文書を先にアクセスし，レコードの抽出を試みる．選択される文書を優先的に処理することで処理コストを削減すると同時に，それらの文書からユーザが求められているレコードを高い抽出精度で獲得する．

4.2 システム概要

図 2 をもとに，提案するシステムの構成について述べる．ここでは例として，ユーザは IT 関係の会社と所在地の情報を抽出したいとする．初期の知識として，最初に**例示レコード集合**がユーザにより与えられる．例示レコード集合はユーザが着目しているトピックを反映している．

文書リポジトリには大量のテキスト文書が格納されている．本研究では，大量のプレーンテキスト文書からなる文書リポジトリを対象としており，ニュース記事を利用する．なお，文書リポジトリは索引付けされており，検索条件が与えられると，対応する文書はランク付けされて返されるものとする．

レコード抽出用文書集合は，文書リポジトリから取り出した，抽出に役立つと考えられる一部の文書からなる．初期のレコード抽出用文書集合は例示レコードを用いて文書リポジトリから選択する．その後，以下で述べるように動的に拡張する．

レコード抽出モジュールは，文書リポジトリの全文

☆ ここでは理解しやすさと簡潔さのため，簡略化を行っている．実際の **Snowball** は，パターン生成のためにより複雑な処理を行う．

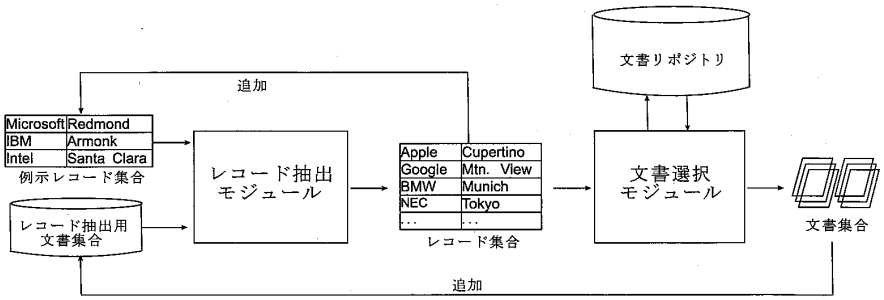


図 2 システム構成
Fig. 2 System architecture.

書ではなく，その一部であるレコード抽出用文書集合の中の文書を解析し，新たなレコードを抽出する．この処理において，ブートストラッピング型の抽出手法（アルゴリズム 1）を利用する．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，パターンをもとにレコード抽出用文書集合から新たなレコードの抽出を図る．

文書選択モジュールは，文書リポジトリから今後のレコード抽出用の文書を選択する．選ばれた文書がレコード抽出用文書集合に追加され，次の抽出処理の対象になる．特定のトピックに関連するレコードがトピックに関連する文書に含まれる可能性が高いと考えられるため，それらの文書を選択し，抽出対象として優先的に利用する．4.3 節では，文書選択手法の詳細について説明する．

文書選択処理を組み合わせたレコード抽出の処理過程を**アルゴリズム 2** にまとめる．レコード抽出処理（6 行目～12 行目）は 3 章で述べたアルゴリズム 1 と同様である．文書選択処理の組み込み（13 行目）と選択された文書に対して優先的に固有表現抽出（14 行目）を行う点が主な違いである．内側のループの終了条件（12 行目）としては，レコード抽出用文書集合において，レコードが抽出できなくなることが例としてあげられる．外側のループの終了条件（15 行目）としては，ユーザが満足する時点で処理を止めることが考えられる．たとえば，求められる数のレコードが達成されると，処理を中止することなどがあげられる．

4.3 文書選択

この節では，文書選択処理の手法について述べる．レコード抽出用文書の選択を工夫しない手法を Baseline 法とする．文書リポジトリは索引付けされているため，文書選択処理について検索条件の作成に焦点を当てる．比較するために，レコードを検索条件とする 2 つの手法（Record without Feedback，Record with Feedback）と，特徴語を検索条件とする 2 つの手法（Salient Word with Feedback，Salient Word without Feedback）の，計 4 つの文書選択手法を提案し，それぞれについて述べる．各文書選択手法の効果について，5 章の実験で評価する．

Algorithm 2 文書選択を融合したレコード抽出

1: *Seed* : 例示レコード集合
2: *Doc* : 文書リポジトリ
3: D : レコード抽出用文書集合，初期時点では例示レコードを含む文書からなる
4: $D\_tag = attach\_tag(D)$
5: **repeat**
6: **repeat**
7: $Occ = find\_occurrences(D\_tag, Seed)$
8: $Pat = generate\_patterns(Occ)$
9: $Rec = extract\_records(D\_tag, Pat)$
10: $Sorted\_Rec = sort\_records(Rec)$
11: $Seed = Seed \cup Top\_k(Sorted\_Rec)$
12: **until** *termination criterion* 1
13: $D\_new = select\_documents(Doc)$
14: $D\_tag = D\_tag \cup attach\_tag(D\_new)$
15: **until** *termination criterion* 2
16: **return** *Rec*

4 つの手法のうち 2 つは，抽出レコードに対するユーザのフィードバックを用いる手法（Record with Feedback，Salient Word with Feedback）であるため，まず次項でユーザのフィードバックの種類について議論する．

4.3.1 ユーザのフィードバック

IT 企業の名称とその所在地を抽出するタスクと想定した場合，ユーザのフィードバックとして下記のような 5 種類が考えられる：

- **正解**：レコードはノイズがなく，正しい対応関係および関連したトピックを持つ．たとえば，(*Apple*, *Cupertino*) というレコードは IT 関係の会社 "*Apple*" と所在地 "*Cupertino*" の正確な対応関係であるため，「正解」と評価する．

- **間違ったトピック**：抽出されたレコードは有効な会社と場所であり，しかも対応関係が正しいペアであるが，ユーザの関心があるトピックのものではない．たとえば，(*BMW*, *Munich*) のペアが抽出されても，IT 情報に興味を持つユーザには，期待されていないものであるため，「間違ったトピック」とマークする．
- **間違った固有表現抽出**：抽出されたレコードにおける会社名と場所が有効でないものもある．たとえば，(*Com Corp.*, *Santa Clara*) や (*Cupertino*, *Calif.*) という，固有表現抽出器で間違って認識されたものが実験結果に見られた．(*Com Corp.*, *Santa Clara*) のペアにある会社名 "*Com Corp.*" は，固有表現抽出器によって "3*Com Corp.*" の一部が切り取られたものである．また，(*Cupertino*, *Calif.*) では，都市名である "*Cupertino*" が会社名に誤認されていることもある．
- **間違った関係**：レコードの会社名と場所は有効なものであるが，会社と場所の関係が間違っている．つまり，場所はその会社の所在地でない．
- **保留**：ユーザは抽出されたレコードが正しいかどうかを判断できない．

後述する実験では，適切なランク付け手法を利用することで，上位にランク付けされたレコードに「間違った固有表現抽出」と「間違った関係」といったノイズ（以下，△と表記）があるものがほとんど現れなかった．そこで，本稿ではこの 2 種類のフィードバックについては特に議論せず，主に「正解」（以下，○と表記）と「間違ったトピック」（以下，×と表記）のフィードバックに着目する．

Record with Feedback と Salient Word with Feedback 文書選択手法では，フィードバックの結果を，ユーザが求めているレコードが含まれる可能性が高い文書の選択に利用する．詳細は次項で述べる．

4.3.2 文書選択手法

- **Baseline（BL）**：文書の選択を工夫せず，文書リポジトリからランダムに抽出対象文書を選ぶ手法である．
- **Record without Feedback（R-F）**：上位にランク付けされたレコードを検索条件とし，文書リポジトリから文書を選択する．具体的には，検索条件はレコードに出現した単語の論理和からなる．**図 3** にある抽出の例では，検索条件は "(*Apple* AND *Cupertino*) OR (*Google* AND *Mtn.* AND *View*) OR (*BMW* AND *Munich*) OR (*NEC* AND *Tokyo*)" になる．

Company	Location
Apple	Cupertino
Google	Mtn. View
BMW	Munich
NEC	Tokyo
⋮	⋮

図 3 抽出例
Fig. 3 Extraction example.

Company	Location	Feedback
Apple	Cupertino	○
Google	Mtn. View	○
BMW	Munich	×
NEC	Tokyo	○
⋮	⋮	⋮

図 4 ユーザフィードバック
Fig. 4 User feedback.

R-F 法で検索条件として利用されるレコードには，異なるトピックのものも存在しうる．ユーザの興味と異なるトピックのレコード（たとえば，(*BMW*, *Munich*) のペア）がランク順の上位に来る場合，そのレコードを検索条件とすると，トピックに不適合な文書が返され，逐次的に違うトピックのレコードが抽出されることが考えられる．

- **Record with Feedback（R+F）**：R-F 法の問題に対して，R+F 法では，ユーザのフィードバックを導入して，異なるトピックのレコードを検索条件から排除する．ユーザはランク付けされたレコードの上位のものに対して，着目するトピックに関連するかどうかを判定する．「正解」と判断されたレコードに出現した単語のみを検索条件とする．**図 4** の例の判断結果では，検索条件は "(*Apple* AND *Cupertino*) OR (*Google* AND *Mtn.* AND *View*) OR (*NEC* AND *Tokyo*)" であり，(*BMW*, *Munich*) が含まれないことになる．
  しかし，抽出処理に時間がかかるため，新たなレコードが出力されるまで，ユーザはしばらく待機する必要がある．さらに，新たなレコードが生成され，ユーザがフィードバックを与えて文書選択処理を再開した後にも，また待機する必要がある．この意味で，この手法は現実的には実行可能なアプローチではない．本稿では，人手でチェックするプロセスを導入することにより，どこまで高い抽出精度を達成できるかという上限を示す目的で，この手法を比較対象に含める．

- **Salient Word with Feedback (SW+F)**：R-F 法と R+F 法はレコードに出現した単語そのものを検索条件とするのに対し，SW+F 法は着目するトピックを表す特徴となる単語を学習し，検索条件を構築する．特徴語の識別は，初期のレコード抽出用文書集合においてレコード抽出処理が終わった後に行われる．この手法では初期のレコード抽出用文書集合からの抽出レコードの結果に対して，ユーザがフィードバックをかける．次に，フィードバックの情報をもとに，適合文書と非適合文書からなる学習文書集合を決める．さらに，学習文書集合の内容を分析することで，適合文書中に出現する単語に順位付けを行う．上位にランク付けされる単語は，ユーザが興味を持つトピックを表す傾向がある．最後に，ランク順に上位の単語を選び，論理和をとって検索条件を作成し，以降の抽出処理用の文書を取得する．SW+F 法は適合文書と非適合文書を決定する際のみ，ユーザのフィードバックの情報を利用する．それ以降の文書の取得では，識別された特徴語を用い，ユーザのフィードバックを必要としない．

  処理の詳細は以下のようになる．

  (1) 適合文書と非適合文書の選択

  まず，1 つ以上のレコードが抽出された文書を候補適合文書とする．候補適合文書集合内の各文書に下記の式でスコアを付与する．

$$score(d) = \frac{r + p \times u}{r + w + u} \times \log(r + p \times u + 1) \quad (1)$$

  d は候補適合文書集合中の文書である．r は文書 d から抽出された，着目するトピックのレコードの数であり，w は他のトピックのレコードの数である．u はユーザが判断していない（あるいは判断できない）レコードの数を表す．p は，ユーザからの判断が与えられていないレコードが，着目するトピックに合致している確率である．異なるタスクに関しては，p に異なる値を経験的に与えることを想定する．後述の実験では p の値を 1/2 に設定した．上述の計算式で計算されたスコアの高い n_1 件の文書を適合文書にする．これで，適合文書は望まれるトピックのレコードが多く抽出され，しかもそのようなレコードの割合が高いものとなる．また，適合文書集合と重ならない文書をランダムに n_2 件を選び，非適合文書とする．

  (2) 特徴語の選択

  Okapi[12] において単語の重み付けに用いられた手法を利用して，(1) で選択された適合文書に出現した各単語 t に下記のスコアを与える：

$$score(t) = \frac{(r_t + 0.5)/(n_1 - r_t + 0.5)}{(n_t - r_t + 0.5)/(n_2 - n_t + r_t + 0.5)} \quad (2)$$

  r_t は単語 t を含む既知の適合文書数，n_t は単語 t を含む文書数，n_1 は既知の適合文書数，n_2 は既存の非適合文書数である．直感的には，単語 t は多くの適合文書に出現し，あまり非適合文書に出現しないほどスコアが高くなるという性質がある．上述のスコアを計算する式に基づいて，適合文書に出現した単語をランク付けする．上位 k 個の単語の論理和をとり，検索条件を作成する．この検索条件で取り出した文書はユーザが着目しているトピックのレコードが含まれる可能性が高いと考えられる．

- **Salient Word without Feedback (SW-F)**：SW-F 法は，ユーザのフィードバックを不要とし，適合文書を選ぶ処理を簡略化した手法である．初期のレコード抽出用文書から抽出されたレコードを人手によって評価することを省き，単純にレコードを抽出できた文書（SW+F 法の候補適合文書）を適合文書とする．以降の特徴語の識別に関する処理は，SW+F 法と同様とする．

5. 実　　験

5.1 実験環境と実験対象

実験に利用した文書リポジトリは 1986 年から 1992 年まで 173,039 件の Wall Street Journal ニュース記事である．全文検索システム Namazu☆を使って，文書リポジトリに索引を付ける．Namazu はブール検索モデルをサポートしており，検索質問に含まれる各キーワードごとに tf-idf[13] の値を求め，それらを単純に足し合わせることによって適合度を計算する．固有表現タグの認識に，University of Illinois が公開している固有表現抽出器☆☆を利用する．英文の文章から人名 (PER)，地名 (LOC)，組織名 (ORG) とその他 (MISC) という 4 種類のエンティティを識別できる．

実験対象として，IT 関係の会社と所在地の抽出と，石油会社と所在地の抽出について検証を行った．最初にユーザによって提供された例示レコード集合はそれぞれ 5 個のペア（**図 5** と**図 6**）からなった．

☆ http://www.namazu.org/

☆☆ http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=ne

IT Company	Location
Apple	Cupertino
Compaq	Houston
Intel	Santa Clara
Sun	Mountain View
Xerox	Stamford

図 5 「IT 会社」対象の例示レコード集合
Fig. 5 Example records for "IT company" target.

Petroleum Company	Location
Blue Tee	New York
Chevron	San Francisco
Conoco	Houston
Halliburton	Dallas
Shell Oil	Houston

図 6 「石油会社」対象の例示レコード集合
Fig. 6 Example records for "Petroleum company" target.

5.2 予備実験

3 章で述べたように，レコード抽出の過程において，ノイズが少なく質の良いレコードが上位に来るよう，レコードのランク付けを行う．レコードへのランク付け手法として，下記のような 4 つの手法を実装し比較した．

- **Pattern number (PN)**：レコードの抽出に用いた抽出パターン数を用いて，レコードをランク付けする手法である．一般的に複数のパターンで抽出されたレコードは，単一のパターンで抽出されたレコードより信頼できると考えられるため，抽出パターン数の降順でレコードをソートする．
- **Link analysis of pattern-record graph (LAPR)**：レコードと抽出パターンからなる 2 部グラフのリンク解析により，レコードをランク付けする手法である．レコードとそのレコードの抽出に用いられたパターンとの関係を一種のリンクと考え，レコードと抽出パターンをノードとし，2 部グラフを作る．良いレコードは多くの良いパターンで抽出されており，良いパターンは多くの良いレコードを抽出するという相互補強の考え方に基づき，レコードとパターンとの 2 部グラフのリンク解析処理[14] を行い，レコードとパターンのスコアを計算する．
- **Document number (DN)**：レコードの出現した文書数を用いて，レコードをランク付けする手法である．数多くの文書から抽出されたレコードは出現回数の少ないレコードより信頼度が高いと考えられる．そこで，レコードに対して出現した文書数の降順でソートする．
- **Link analysis of document-record graph (LADR)**：レコードと文書からなる 2 部グラフのリンク解析により，レコードをランク付けする手法である．LAPR 法と同様に，レコードとレコードの出現した文書との関係を 2 部グラフで表し，リンク解析処理でレコードと文書のスコアを計算する．

予備実験として，4 つのランク付け手法の比較を行った．詳細を付録 A.1 に示す．要約すると，出現文書数が多いことを重視する DN 法と LADR 法よりも，抽出パターン数が多いことを重視する PN 法と LAPR 法が，ノイズの面で優れていた．

PN 法と LAPR 法に関しては，PN 法よりも LAPR 法の方がいくぶん優れていたが，その差は顕著ではない．今回の実験対象については，2 部グラフのノード数が少なく，ノイズも含まれるため，LAPR 法のリンク解析のアプローチが PN 法に大きく差をつけるに至らなかったと考えられる．一方，LAPR 法では繰返し処理によるスコアの計算が必要であり，処理時間が大きいという問題がある．そこで，以降の実験では，レコードのランク付け手法として PN 法を用いる．なお，PN 法では，あるレコードに対して抽出パターン数が同じである場合，そのレコードが出現した文書数を用いてソートする．

5.3 実験パラメータの設定

4.3.2 項で述べた SW+F 法と SW-F 法の実験パラメータの設定について述べる．IT 会社を対象とした場合，SW+F 法で特徴語を識別する際には，適合文書数と非適合文書数を $n_1 = n_2 = 150$ 件とした．SW-F 法の場合，初期のレコード抽出用文書集合（図 5 の例示レコードを含む文書）の中で，レコードを抽出できた $n_1 = 333$ 件の文書を適合文書とし，ランダムに同じく $n_2 = 333$ 件の非適合文書を取り出し，特徴語の識別に用いた．IT 会社の対象においては，SW+F 法と SW-F 法で検索条件として利用した特徴語の数を $k = 50$ 個に限定した．識別された特徴語を**表 1** に示す．なお，括弧内の数字は単語のランク順である．SW+F 法と SW-F 法で識別されたそれぞれの 50 個の単語に共通のもの（表 1 で太字で書かれている単語）が 32 個ある．これらの単語は IT 会社の社名に出現した単語（compaq, dataquest, intel, mips, silicon, sun, xerox），IT 会社の場所に出現した単語（calif, clara, cupertino, maynard, mountain, santa, sunnyvale）と IT のトピックを表す単語（386, 486, chips, clones, computer, computers, computing, desktop, logic, macintosh, mi-

表 1 「IT 会社」を対象とした場合の 50 個の特徴語
Table 1 The 50 salient words for "IT company" target.

(a) SW+F 法	(b) SW-F 法
386(27), **486**(49), apollo(47), **calif**(1), **chips**(43), **clara**(3), clone(39), **clones**(30), **compaq**(29), compatible(19), **computer**(26), **computers**(7), **computing**(18), **cupertino**(6), **dataquest**(28), **desktop**(21), digital(42), edge(46), hewlett(24), houston(45), **logic**(44), **intel**(33), jose(13), **macintosh**(9), **maynard**(50), memory(31), micro(12), **microprocessor**(5), **microprocessors**(17), **microsystems**(4), **mips**(38), models(22), motorola(36), **mountain**(2), networks(34), older(40), packard(23), **pc**(10), **risc**(35), **santa**(37), **silicon**(25), software(41), stamford(20), **sun**(15), **sunnyvale**(11), supplier(48), **unix**(32), **workstation**(14), **workstations**(8), **xerox**(16)	**386**(20), **486**(43), alto(19), apple(7), armonk(32), **calif**(14), **chips**(34), circuits(40), **clara**(13), **clones**(27), **compaq**(5), **computer**(26), **computers**(30), **computing**(17), **cupertino**(2), custom(49), **dataquest**(33), **desktop**(15), hambrecht(44), **intel**(12), **logic**(25), **macintosh**(6), **maynard**(38), **microprocessor**(4), **microprocessors**(11), microsoft(3), **microsystems**(1), milpitas(39), **mips**(31), **mountain**(29), networking(46), palo(18), **pc**(41), printers(48), quist(42), redmond(37), **risc**(35), **santa**(8), seagate(45), **silicon**(21), sparc(47), **sun**(23), **sunnyvale**(16), tasks(50), **unix**(22), wash(24), windows(28), **workstation**(10), **workstations**(36), **xerox**(9)

表 2 「石油会社」を対象とした場合の 30 個の特徴語
Table 2 The 30 salient words for "Petroleum company" target.

(a) SW+F 法	(b) SW-F 法
amoco(10), arco(29), **barrels**(28), **bartlesville**(24), **chevron**(2), coast(7), collapse(30), **conoco**(17), crude(11), **exxon**(8), fields(12), **francisco**(22), fuel(15), gallon(26), halliburton(21), **houston**(16), **mobil**(6), oil(14), **petroleum**(20), plains(23), **pont**(9), refineries(18), **refinery**(3), **refining**(4), richfield(13), **san**(27), santa(25), **shell**(1), texaco(5), unocal(19)	alberta(14), **amoco**(7), **barrels**(22), **bartlesville**(23), bottom(20), calgary(18), **chevron**(1), **conoco**(8), drilling(9), du(29), dutch(11), **exxon**(6), **francisco**(13), gasoline(25), gulf(10), **houston**(5), iron(16), **mobil**(3), petro(28), petrochemical(19), **petroleum**(24), pipeline(12), **pont**(30), refiners(17), **refinery**(15), **refining**(2), river(21), royal(246), **san**(27), **shell**(4)

croprocessor, microprocessors, microsystems, pc, risc, unix, workstation, workstations) の 3 種類からなる．共通しない単語も主にこの 3 種類の単語である．

石油会社を対象とした場合，SW+F 法と SW-F 法では，適合文書数と非適合文書数をそれぞれ $n_1 = n_2 = 150$ と $n_1 = n_2 = 182$ 件とした．石油会社に関する文書数は IT 会社に関する文書数より少ないと考えられるため，石油会社の実験で取得する文書数は，IT 会社の場合より小さく限定した．石油会社を対象とした場合では，より少ない文書を抽出対象としたため，SW+F 法と SW-F 法で利用した特徴語の数をやや低めに $k = 30$ 個とした．**表 2** は石油会社の対象の特徴語を示したものである．これらに共通の単語が 15 個あり，石油会社の社名に現れた単語 (amoco, chevron, conoco, exxon, mobil, pont, shell)，場所に現れた単語 (bartlesville, francisco, houston, san) および石油会社に関する記事によく出る単語 (barrels, petroleum, refinery, refining) からなる．

表 1 と表 2 で示される単語は，多くの IT または石油会社に関する文書に出現し，あまり他のトピックの文書に出現しないものであるため，提案したスコア付け方式 (4.3.2 項の式 (2)) が妥当であると考えられる．

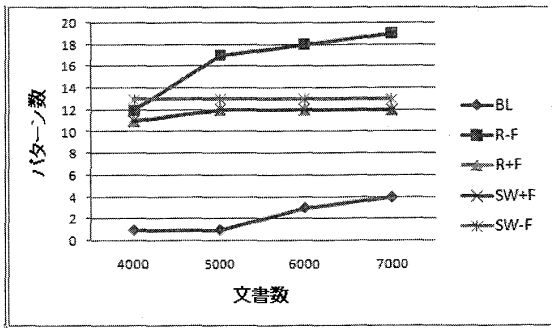


図 7 「IT 会社」対象のパターン数
Fig. 7 The numbers of patterns for "IT company" target.

5.4 抽出結果

IT 会社の対象について，4.3.2 項で述べた各文書選択手法で取得する文書数を 4,000 件から 7,000 件まで 1,000 件ごとに変動させて，抽出パターン数と抽出レコード数を調べたものを**図 7** と**図 8** に示す．石油会社を対象とした場合，石油会社に関する文書数は IT 会社に関する文書数より少ないため，2,000 件から 5,000 件までのより少ない文書数に限定した．石油会社を対象とした場合の抽出パターン数と抽出レコード数を**図 9** と**図 10** に示す．文書の選択を工夫しない BL 法では，取り出した文書から獲得されたパターン数とレコード数はきわめて少ない．IT 会社の場合，

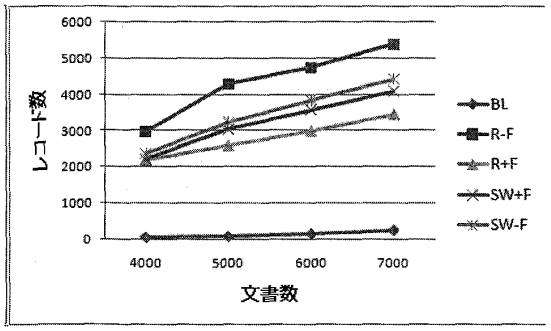


図 8 「IT 会社」対象のレコード数
Fig. 8 The numbers of records for "IT company" target.

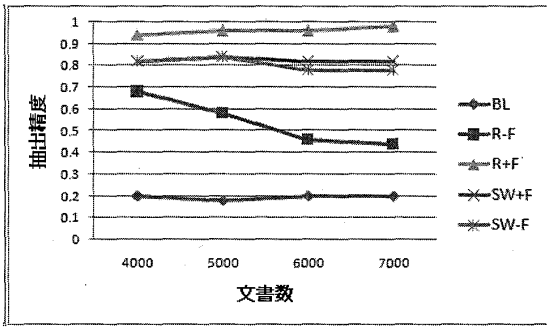


図 11 「IT 会社」対象の抽出精度
Fig. 11 Extraction accuracy for "IT company" target.

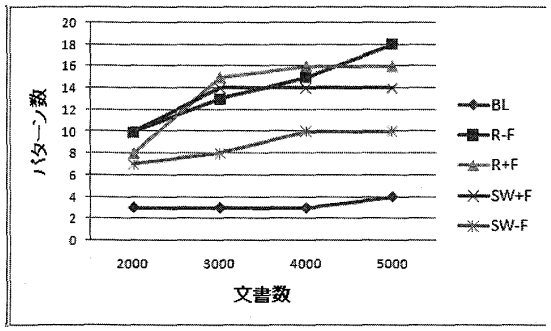


図 9 「石油会社」対象のパターン数
Fig. 9 The numbers of patterns for "Petroleum company" target.

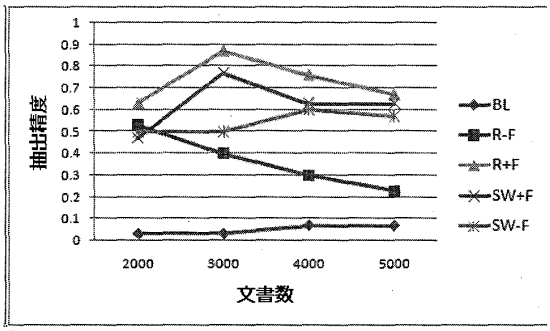


図 12 「石油会社」対象の抽出精度
Fig. 12 Extraction accuracy for "Petroleum company" target.

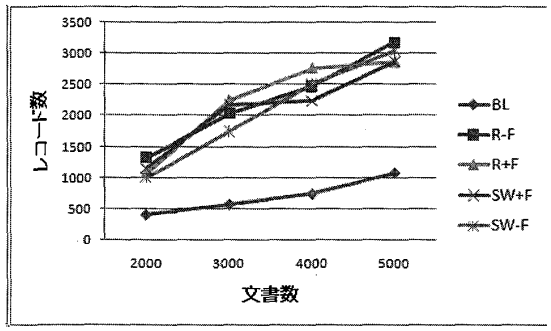


図 10 「石油会社」対象のレコード数
Fig. 10 The numbers of records for "Petroleum company" target.

7,000 件の文書を取得するまで，4 個のパターンしか生成されず，263 個のレコードしか抽出されなかった．石油会社の場合，5,000 件の文書を取得した時点のパターン数とレコード数はそれぞれわずか 4 個と 1,075 個であった．一方，他の 4 つの手法では，はるかに多くのパターンとレコードが得られた．IT 会社の対象で 7,000 件の文書を取得した時点のパターン数は 12 個～19 個程度であり，レコード数は 3,458 個～5,390 個程度であった．石油会社の対象で 5,000 件の文書を取得した時点のパターン数とレコード数は，それぞれ 10 個～18 個程度と 2,870 個～3,178 個程度であった．これは文書選択処理を組み込むことで，より多くのパターンとレコードを獲得できることを示している．

1 つの例として，SW-F 法で 7,000 件の文書を取得した時点の抽出パターンと抽出レコードの様子を付録 A.2 に示す．

5.5 抽出精度の評価

図 11 と**図 12** は IT 会社と石油会社をそれぞれ対象とした場合の抽出精度と文書数の関係を表すものである．抽出精度は，抽出したレコードを 5.2 節で述べた PN ランク付け手法に基づいてソートした上位 m 個のうちで，ノイズがなく着目のトピックに適合するレコードの数の割合とする．IT 会社の対象では $m = 50$ とした．石油会社については，石油会社と所在地のレコード数は IT 会社と所在地のレコード数より少ないと考えられるため，$m = 30$ とした．なお，レコードにノイズがあるかどうかや，着目するトピックに適合するかどうかへの判断は著者により行った．ノイズについては，IT 会社を対象とした場合，各文書選択手法，各文書数とも，PN 法で上位 50 個にソートされたレコードのうち，ノイズのあるものは平均 2～3 個程度で，最大が 7 個で，最小が 0 個であった．石油会社の場合の上位 30 個のレコードのうち，ノイズのあるものは平均 1～2 個程度で，最大が 10 個で，最小が 0 個であった．

図 11 と図 12 に注目すると，BL 法が最も抽出精度が低い．IT 会社と石油会社を対象とした場合の精度

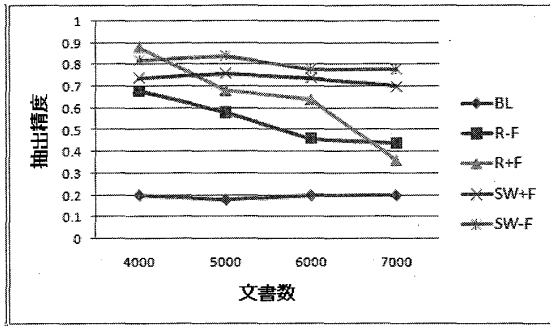


図 13 「IT 会社」対象の抽出精度
(ユーザがフィードバックしたレコードを除く場合)
Fig. 13 Extraction accuracy for "IT company" target (excluding the feedback records).

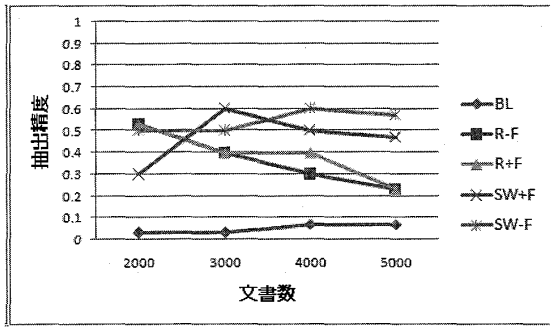


図 14 「石油会社」対象の抽出精度
(ユーザがフィードバックしたレコードを除く場合)
Fig. 14 Extraction accuracy for "Petroleum company" target (excluding the feedback records).

がそれぞれ約 0.2，約 0.05 であった．R-F 法は文書取得件数が少ないときに良い精度が得られたが，文書数が増えるとともに精度が落ちていく．IT 会社を対象とした場合，取得する文書の件数が 4,000 件であるときの抽出精度が 0.68 であったが，7,000 件の文書を取得した時点における抽出精度が 0.44 に下がった．石油会社の対象についても，同様の下降傾向が見られた．R+F 法は人手でのチェックにより最も高い精度を獲得している．IT 会社を対象とした場合の抽出精度は約 0.95 であり，石油会社を対象とした場合の抽出精度は 0.63〜0.87 程度であった．少数のフィードバックしか必要としない SW+F 法とフィードバックを必要としない SW-F 法の抽出精度は，R+F 法の次に続いている．それらの精度は R+F 法より 10%ほど低い精度であり，それほど差がないことが分かる．これらのことから，特徴語を利用する SW+F 法と SW-F 法は，実用的であり抽出精度も優れているといえる．

図 11 と図 12 では，ユーザがフィードバックをしたレコードと，フィードバックをしていないレコードを区別せず扱っている．この場合，SW+F 法と SW-F 法を比較すると，フィードバックを利用した SW+F 法でやや高い精度が得られている．一方，抽出結果から，ユーザがすでにフィードバックをしたレコードを除いて抽出精度を比較したものを，**図 13** と**図 14** に示す．この場合，BL 法，R-F 法および SW-F 法はユーザのフィードバックを必要としないため，抽出精度の値が不変である．一方，R+F 法と SW+F 法の抽出精度は低くくなり，最終的に SW-F 法より下回っている．SW+F 法は，最初の段階でのみ，ユーザのフィードバックを利用している．そのため，異なる数の文書を取得する場合でも，必要とするフィードバック数は同じである．IT 会社に関する実験では，最初に行ったフィードバック数は 20 回であった．図 13 と図 11 を比較すると，フィードバックされた 20 個のレコードを評価対象から除いた結果として，SW+F 法の抽出精度が 8%ほど低くなっていることが分かる．石油会社に関する実験では，最初に行ったフィードバック数は同じく 20 回であった．図 14 と図 12 を比較すると，SW+F 法の抽出精度は 15%ほど低下した．R+F 法については，取得する文書件数が増えるにつれ，必要とするフィードバック数も増加する．たとえば，IT 会社を対象とした場合，4,000〜7,000 件の文書を取得した各時点のフィードバック数はそれぞれ 86，116，116，134 であった．取得文書件数が大きくなるほど，抽出精度が下がる程度がより大きくなる．ユーザがフィードバックをしたより多くのレコードが評価対象から除かれるためである．石油会社の対象についても，同様な傾向が見られた．

また，図 11 と図 12 を比べると，石油会社を対象とした場合の抽出精度は，IT 会社の場合よりも全般的に低くなっていることが分かる（図 13 と図 14 も同様である）．IT 会社を対象とした場合に達成できた 0.98 の最大精度（図 11 の R+F 法で 7,000 件文書を取得した時点）に対して，石油会社を対象とした場合の精度の最大値（図 12 の R+F 法で 3,000 件文書を取得した時点）は 0.87 であった．石油会社に関する文書数は IT 会社に関する文書数より少ないため，石油会社を対象とした文書を取得する際，異なるトピックの文書が混在しやすいことが理由として考えられる．

5.6 処理コストの評価

処理コストについては，指定された数のレコードが抽出された時点における，各文書選択手法で処理した文書数，処理時間，およびフィードバック数で評価する．IT 会社を対象とし，レコード数を 3,000 個に限定した場合の処理コストについての実験結果を**表 3** に示す．なお，表 3 ではその時点の抽出精度も提示する．同じ数のレコードを獲得するため，BL 法では，ほかの文書選択手法より約 2〜3 倍の文書（12,000 件）を

表 3 IT 会社を対象とした 3,000 個のレコードを抽出するまでの処理コスト
Table 3 Processing cost until 3,000 records are extracted for "IT company" target.

	処理文書数	処理時間	フィードバック数	抽出精度
Baseline (BL)	12,000	11.38 h	0	0.14
Record without Feedback (R-F)	4,100	3.95 h	0	0.66
Record with Feedback (R+F)	6,000	5.70 h	116	0.96
Salient Word with Feedback (SW+F)	5,000	4.81 h	20	0.84
Salient Word without Feedback (SW-F)	4,800	4.69 h	0	0.82

表 4 単一または複数のレコードを抽出できた文書数の統計
Table 4 The analysis of the numbers of documents with singular or plural records extracted.

	レコードを抽出できた文書総数	1 つのレコードのみを抽出できた文書数（割合）	複数のレコードを抽出できた文書数（割合）
Baseline (BL)	251	229 (91%)	22 (9%)
Record without Feedback (R-F)	4,448	2,264 (51%)	2,184 (49%)
Record with Feedback (R+F)	3,749	2,217 (59%)	1,532 (41%)
Salient Word with Feedback (SW+F)	3,310	1,855 (56%)	1,455 (44%)
Salient Word without Feedback (SW-F)	3,340	1,977 (59%)	1,363 (41%)

表 5 単一または複数の文書に出現したレコード数の統計
Table 5 The analysis of the numbers of records extracted from singular or plural documents.

	レコード総数	1 つの文書のみに出現したレコード数（割合）	複数の文書に出現したレコード数（割合）
Baseline (BL)	263	252 (96%)	11 (4%)
Record without Feedback (R-F)	5,390	4,514 (84%)	876 (16%)
Record with Feedback (R+F)	3,458	2,844 (82%)	614 (18%)
Salient Word with Feedback (SW+F)	4,105	3,441 (84%)	664 (16%)
Salient Word without Feedback (SW-F)	4,422	3,719 (84%)	703 (16%)

表 6 単一または複数のパターンで抽出されたレコード数の統計
Table 6 The analysis of the numbers of records extracted by singular or plural patterns.

	レコード総数	1 つのパターンのみで抽出したレコード数（割合）	複数のパターンで抽出したレコード数（割合）
Baseline (BL)	263	257 (98%)	6 (2%)
Record without Feedback (R-F)	5,390	4,968 (92%)	422 (8%)
Record with Feedback (R+F)	3,458	3,176 (92%)	282 (8%)
Salient Word with Feedback (SW+F)	4,105	3,834 (93%)	271 (7%)
Salient Word without Feedback (SW-F)	4,422	4,103 (93%)	319 (7%)

処理する必要があり，約 2〜3 倍の処理時間（約 11 時間）がかかる．R-F 法，R+F 法，SW+F 法および SW-F 法については，同程度の処理文書数（4,000〜6,000 件程度）と処理時間（4〜6 時間程度）が必要である．

BL 法以外の 4 つの手法を比較すると，R-F 法の処理時間が比較的に短いものの，抽出精度が低い．また，図 11 と図 12 で示されるように，取得する文書数が増えるとともに，R-F 法の抽出精度がさらに悪くなる．R+F 法は，処理文書数が少なく，処理時間が短いというこの実験の設定のもとでは高い精度が達成できた．しかし，この手法は多くのユーザフィードバックを要する．少数のフィードバックしか必要としない SW+F 法と，フィードバックを必要としない SW-F 法は，少ない処理文書数，短い処理時間で，比較的高い抽出精度を得ている．これらの 2 つの手法では，特徴語の識別に余分な計算時間を必要とするが，処理時間が相対的に小さいことが確認できた．具体的には，適合文書と非適合文書をそれぞれ 333 件とした場合，計算時間は 8.5 分ほどである．

5.7 その他の実験結果

この節では，その他の実験結果についての要約を示す．まず，単一または複数のレコードを抽出できた文書数，単一または複数の文書に出現したレコード数および，単一または複数のパターンで抽出されたレコード数の統計情報を**表 4**，**表 5** と**表 6** に示す．なお，これらは IT 会社を対象とし，7,000 件文書を取得した時点の分析結果である．表 4 に着目すると，BL 法以

外の手法について，1つのレコードのみを抽出できた文書と複数のレコードを抽出できた文書が，それぞれレコードを抽出できた文書の5割強と4割弱を占めたことが分かる．表5は抽出されたレコードのうち，8割強のものがただ1つの文書に出現しており，2割弱のものが複数の文書に出現したことを示す．表6から，9割以上のレコードが1つのパターンで抽出されており，複数のパターンで抽出されたレコードの割合が1割未満であることが分かる．

また，IT会社と石油会社以外の対象として，バイオテクノロジ会社と所在地の対象の抽出についても検証を行った．抽出結果，抽出精度と処理コストに関して，IT会社と石油会社の対象と同じ傾向が見られた．実験の詳細は論文15)にある．

6. まとめと今後の展開

本稿では，トピックに適合するレコードを効率的に抽出するために文書選択を融合したレコード抽出システムの仕組みを提案した．文書選択によりレコード抽出の処理コストを削減する点と，レコードのトピックも考慮した抽出精度を向上させる点に新規性がある．

BL，R-F，R+F，SW+F，SW-Fといった5つの文書選択手法を比べ，それぞれの効果について分析した．文書選択を工夫しないBF法では，多大な処理コストがかかるにもかかわらず，良い抽出精度が得られない．R-F法は処理コストの削減に有効であるが，文書取得件数が多い場合，抽出精度を保つことができない．R+F法では最大の抽出精度が得られているが，人手での大量のチェックが不可欠なので，現実的に実行可能ではない．少数のフィードバックしか必要としないSW+F法と，フィードバックを必要としないSW-F法は，実用的なアプローチであり，処理コストの削減と抽出精度の確保の両面で優れている．

今回の実験は会社と所在地のペアに限られたが，今後は他の抽出対象の試みを行いたい．また，文書から抽出されたレコードと既存データベースとの統合も今後の課題の1つである．

謝辞 投稿論文に対し，詳細にご査読いただき価値あるコメントをいただいたメタレビュアおよび査読委員の皆様に感謝いたします．本研究の一部は，文部科学省科学研究費補助金特定領域研究（19024006）ならびに日本学術振興会科学研究費（19300027），科学技術振興機構戦略的創造研究推進事業CRESTの支援による．

参考文献

1) Brin, S.: Extracting Patterns and Relations from the World Wide Web, *Proc. WebDB*, pp.172–183 (1998).
2) Zhang, R.Y., Lakshmanan, L.V.S. and Zamar, R.H.: Extracting Relational Data from HTML Repositories, *SIGKDD Explorations*, Vol.6, No.2, pp.5–13 (2004).
3) Agichtein, E. and Gravano, L.: Snowball: Extracting Relations from Large Plain-Text Collections, *Proc. ACM DL*, pp.85–94 (2000).
4) 楠村幸貴，土方嘉徳，西田正吾：レイアウト構造の解析とテンプレートの交叉による教師無し情報抽出手法，電子情報通信学会第二種研究会資料，WI2-2007-02, pp.7–12 (2007).
5) Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. and Slattery, S.: Learning to Construct Knowledge Bases from the World Wide Web, *Artificial Intelligence*, Vol.118, No.1-2, pp.69–113 (2000).
6) Baumgartner, R., Flesca, S. and Gottlob, G.: Visual Web Information Extraction with Lixto, *Proc. VLDB*, pp.119–128 (2001).
7) Kushmerick, N.: Wrapper Induction: Efficiency And Expressiveness, *Artificial Intelligence*, Vol.118, No.1-2, pp.15–68 (2000).
8) Chakrabarti, S., van den Berg, M. and Dom, B.: Focused Crawling: A New Approach to Topic-specific Web Resource Discovery, *Computer Networks*, Vol.31, No.11-16, pp.1623–1640 (1999).
9) Chakrabarti, S., Punera, K. and Subramanyam, M.: Accelerated Focused Crawling through Online Relevance Feedback, *Proc. WWW*, pp.148–159 (2002).
10) Agichtein, E. and Gravano, L.: Querying Text Databases for Efficient Information Extraction, *Proc. ICDE*, pp.113–124 (2003).
11) Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P. and Vilain, M.: Mixed-Initiative Development of Language Processing Systems, *Proc. ANLP*, pp.348–355 (1997).
12) Robertson, S.E.: Overview of the Okapi projects, *Journal of the American Society for Information Science*, Vol.53, No.1, pp.3–7 (1997).
13) Salton, G. and McGill, M.: *Introduction to Modern Information Retrieval*, McGraw-Hill (1983).
14) Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment, *J. ACM*, Vol.46, No.5, pp.604–632 (1999).

15) Zhang, J., Ishikawa, Y. and Kitagawa, H.: Record Extraction Based on User Feedback and Document Selection, *Proc. APWeb/WAIM*, pp.574–585 (2007).

付　　録

A.1 各レコードランク付け手法のランク順に上位50個の結果

IT会社とその所在地を抽出対象とし，SW-F文書選択手法で取り出した7,000件の文書から抽出されたレコードを用いて，各ランク付け手法（5.2節）の効果について調べた．ただし，ランク付けの目的はノイズのないレコードを見つけることであるため，ここではレコードにノイズがあるかどうかのみを評価し，レコードがトピックに合致するかどうかを考慮しないとする．各ランク付け手法で上位50個にソートされたレコードの結果を**図15**，**図16**，**図17**および**図18**に示す．順位に♣記号を付けたレコードはノイズのあるものである．

図17と図18で示されるように，DN法とLADR法で上位にランク付けされたレコードの中には，「間違った固有表現抽出」のもの（(*Milpitas*, *Calif.*)のようなペア）が多数見られた．これは固有表現抽出器のミスで地名である“*Milpitas*”が組織名として認識されたことで，（地名，地名）のペアが抽出されたこと

順位	会社名	場所
1	Sun Microsystems Inc.	Mountain View
2	Dataquest Inc.	San Jose
3	Intel	Santa Clara
4	Sun	Mountain View
5	Hewlett-Packard	Palo Alto
6	IBM	Armonk
7	Apple	Cupertino
8	Intel Corp.	Santa Clara
9	Tandem Computers Inc.	Cupertino
10	Motorola	Schaumburg
11	Syntex Corp.	Palo Alto
12	Syntex	Palo Alto
13	Feshbach Brothers	Palo Alto
14	First Interstate Bancorp	Los Angeles
15	NEC Corp.	Japan
16	Toshiba Corp.	Japan
17♣	Quist Inc.	San Francisco
18	Borland International Inc.	Scotts Valley
19	Hewlett-Packard Co.	Palo Alto
20	Canon Inc.	Japan
21	Silicon Graphics Inc.	Mountain View
22	Cypress Semiconductor Corp.	San Jose
23	Mips	Sunnyvale
24	Software Publishing Corp.	Mountain View
25	Ardent Computer Corp.	Sunnyvale
26	Lotus	Cambridge
27	Institutional Venture Partners	Menlo Park
28	Metaphor	Mountain View
29	Benham Capital Management	Palo Alto
30	Acer Group	Taiwan
31	Conner Peripherals Inc.	San Jose
32	Fujitsu Ltd.	Japan
33	Advanced Micro Devices Inc.	Sunnyvale
34♣	Digital	Maynard
35	Microsoft	Redmond
36	Xerox	Stamford
37	Hitachi Ltd.	Japan
38	National Semiconductor Corp.	Santa Clara
39	Advest Inc.	Hartford
40	Forrester Research Inc.	Cambridge
41	Apple Computer Inc.	Cupertino
42♣	Com Corp.	Santa Clara
43	Yankee Group	Boston
44	Robinson-Humphrey Co.	Atlanta
45	Siemens	West Germany
46	Dain Bosworth Inc.	Minneapolis
47	International Computers Ltd.	Britain
48	Mips Computer Systems Inc.	Sunnyvale
49	Carnegie Mellon University	Pittsburgh
50	Fujitsu	Japan

図15 PN法によるレコードのランク付け結果
Fig. 15 The result of records ranked by the PN method.

順位	会社名	場所
1	Sun Microsystems Inc.	Mountain View
2	Dataquest Inc.	San Jose
3	Syntex Corp.	Palo Alto
4	Feshbach Brothers	Palo Alto
5	Benham Capital Management	Palo Alto
6	NEC Corp.	Japan
7	Hewlett-Packard	Palo Alto
8	First Interstate Bancorp	Los Angeles
9	Tandem Computers Inc.	Cupertino
10	Syntex	Palo Alto
11	Intel Corp.	Santa Clara
12	Software Publishing Corp.	Mountain View
13	Ardent Computer Corp.	Sunnyvale
14	Conner Peripherals Inc.	San Jose
15	Cypress Semiconductor Corp.	San Jose
16	Borland International Inc.	Scotts Valley
17	Acer Group	Taiwan
18♣	Quist Inc.	San Francisco
19	Forrester Research Inc.	Cambridge
20	Geduld Inc.	New York
21	Siemens	Germany
22	Eastman Kodak Co.	Rochester
23	Cray Research Inc.	Minneapolis
24	Dain Bosworth Inc.	Minneapolis
25	SoundView Financial Group	Stamford
26	Intel	Santa Clara
27	Toshiba Corp.	Japan
28	Geduld	New York
29	Alza Corp.	Palo Alto
30	Bankers Trust Co.	New York
31	Yankee Group	Boston
32	Sun	Mountain View
33	MIPS Computer Systems Inc.	Sunnyvale
34♣	Com Corp.	Santa Clara
35	Varian Associates Inc.	Palo Alto
36	Cooper Cos.	Palo Alto
37	Samsung Electronics Co.	Korea
38	Carnegie Mellon University	Pittsburgh
39	Robinson-Humphrey Co.	Atlanta
40	Fujitsu Ltd.	Japan
41	International Computers Ltd.	Britain
42	Hanson PLC	Britain
43	Bridgestone	Japan
44	Fujitsu	Japan
45	Electronic Arts	San Mateo
46	Hewlett-Packard Co.	Palo Alto
47	Metaphor	Mountain View
48	Symantec	Cupertino
49	Advest Inc.	Hartford
50	Apple Computer Inc.	Cupertino

図16 LAPR法によるレコードのランク付け結果
Fig. 16 The result of records ranked by the LAPR method.

順位	会社名	場所
1♣	Milpitas	Calif.
2	IBM	Armonk
3♣	Redmond	Wash.
4♣	Cambridge	Mass.
5♣	Armonk	N.Y.
6♣	Stamford	Conn.
7	Apple	Cupertino
8♣	Cupertino	Calif.
9♣	Digital	Maynard
10♣	Chelmsford	Mass.
11	Microsoft	Redmond
12	Xerox	Stamford
13♣	Emeryville	Calif.
14♣	Atchison	Topeka
15♣	Princeton	N.J.
16	Motorola	Schaumburg
17♣	Waltham	Mass.
18♣	Nashville	Tenn.
19♣	Hampton	N.H.
20♣	Framingham	Mass.
21♣	Bellevue	Wash.
22	Nintendo	America
23♣	Natick	Mass.
24♣	Minnetonka	Minn.
25♣	Holliston	Mass.
26♣	Watertown	Mass.
27	Mips	Sunnyvale
28♣	Kent	Wash.
29	Intel	Santa Clara
30	Hewlett-Packard	Palo Alto
31	Compaq	Houston
32	Xoma	Berkeley
33	Syntex	Palo Alto
34	Sun	Mountain View
35♣	Marysville	Ohio
36♣	Marlborough	Mass.
37♣	Hartford	Conn.
38♣	Buffalo	N.Y.
39♣	Brooklyn	N.Y.
40♣	Boulder	Colo.
41♣	Belmont	Calif.
42♣	Worthington	Ohio
43♣	Westborough	Mass.
44♣	Troy	Mich.
45♣	Tampa	Fla.
46	Sun	Radnor
47♣	Southfield	Mich.
48♣	Secaucus	N.J.
49♣	Reston	Va.
50♣	Phoenix	Ariz.

図 17 DN 法によるレコードのランク付け結果
Fig. 17 The result of records ranked by the DN method.

になる．このようなペアは正しいレコードではないが，出現した文書数が多いため，DN 法と LADR 法で高く評価された．しかし，このようなレコードの抽出に用いたパターン数は 1 つであり，PN 法（図 15）と LAPR 法（図 16）の評価では上位には来ないことが一般的であった．

PN 法と LAPR 法を比較すると，両方とも上位にランク付けされたレコードがほとんどノイズのないものであることが分かる．具体的には，上位 50 個のレコードのうち，ノイズのあるものがそれぞれわずか 3 個，2 個である．また，32 個のレコードは共通のものである．

順位	会社名	場所
1♣	Milpitas	Calif.
2♣	La Jolla	Calif.
3♣	Redmond	Wash.
4♣	Cambridge	Mass.
5	Intel Corp.	Santa Clara
6♣	Belmont	Calif.
7	Advanced Micro Devices Inc.	Sunnyvale
8♣	Hampton	N.H.
9	NEC Corp.	Japan
10	Cypress Semiconductor Corp.	San Jose
11♣	Atchison	Topeka
12	Intel	Santa Clara
13	Nintendo Co.	Japan
14	Convergent Technologies Inc.	Santa Clara
15	Nintendo	America
16♣	Emeryville	Calif.
17	MIPS Computer Systems Inc.	Sunnyvale
18	Motorola	Schaumburg
19	Ministry of International Trade and Industry	Japan
20	Drexel Burnham Lambert Inc.	New York
21	Borland International Inc.	Scotts Valley
22	Institutional Shareholder Services	Washington
23	Nintendo Ltd.	Japan
24	Dow Jones News Service	New York
25	Bullet-Proof Software	Redmond
26♣	Chelmsford	Mass.
27	Fidelity Investments	Boston
28	Financial Strategic Portfolio Technology Fund	Denver
29	Kemper Technology Fund	Chicago
30♣	Boulder	Colo.
31	Seagate	Scotts Valley
32	The Wall Street Journal	New York
33	First Tennessee National	Memphis
34	OverTheCounter Securities Fund	Fort Washington
35♣	Cypress	Calif.
36	Chugai Pharmaceutical Co.	Japan
37	Commerce Clearinghouse Inc.	Riverwoods
38♣	Nashville	Tenn.
39	Adobe	Mountain View
40	Stock Exchange	London
41	IBM	Armonk
42	Mips	Sunnyvale
43	Fuji Heavy Industries Ltd.	Japan
44♣	Worthington	Ohio
45	Siemens AG	Germany
46	Geduld Inc.	New York
47♣	Waltham	Mass.
48♣	Melville	N.Y.
49	Fujitsu Microelectronics Inc.	U.S.
50♣	Carson	Calif.

図 18 LADR 法によるレコードのランク付け結果
Fig. 18 The result of records ranked by the LADR method.

A.2 抽出パターンと抽出レコードの例

図 19 と**図 20** は IT 会社を対象とし，SW-F 文書選択手法で 7,000 件の文書を取得した時点の抽出結果であり，**図 21** と**図 22** は石油会社を対象とし，SW-F 法で 5,000 件の文書を取得した時点の抽出結果である．なお，「△」と表記されたレコードは「間違った固有表現抽出」または「間違った関係」のものである．「○」は着目するトピックのペアを表し，「×」は他のトピックのものを表している．IT 会社を対象とした場合の上位 50 個のレコードのうち，3 個はノイズのあるも

ID	抽出パターン	抽出されたレコード数
1	ORG , LOC	1,690
2	ORG in LOC	904
3	ORG of LOC	619
4	ORG , a LOC	547
5	LOC 's ORG	430
6	ORG 's LOC	223
7	ORG , based in LOC	197
8	ORG , of LOC	84
9	ORG , the LOC	81
10	ORG , in LOC	39
11	ORG , which is based in LOC	29
12	ORG is based in LOC	17
13	ORG , both of LOC	11

図 19 「IT 会社」対象の抽出パターンおよびそのパターンで抽出されたレコード数

Fig. 19 Extraction patterns and the numbers of records extracted using the extraction patterns for "IT company" target.

順位	会社名	場所	判定
1	Sun Microsystems Inc.	Mountain View	○
2	Dataquest Inc.	San Jose	○
3	Intel	Santa Clara	○
4	Sun	Mountain View	○
5	Hewlett-Packard	Palo Alto	○
6	IBM	Armonk	○
7	Apple	Cupertino	○
8	Intel Corp.	Santa Clara	○
9	Tandem Computers Inc.	Cupertino	○
10	Motorola	Schaumburg	○
11	Syntex Corp.	Palo Alto	○
12	Syntex	Palo Alto	○
13	Feshbach Brothers	Palo Alto	×
14	First Interstate Bancorp	Los Angeles	×
15	NEC Corp.	Japan	○
16	Toshiba Corp.	Japan	○
17	Quist Inc.	San Francisco	△
18	Borland International Inc.	Scotts Valley	○
19	Hewlett-Packard Co.	Palo Alto	○
20	Canon Inc.	Japan	○
21	Silicon Graphics Inc.	Mountain View	○
22	Cypress Semiconductor Corp.	San Jose	○
23	Mips	Sunnyvale	○
24	Software Publishing Corp.	Mountain View	○
25	Ardent Computer Corp.	Sunnyvale	○
26	Lotus	Cambridge	○
27	Institutional Venture Partners	Menlo Park	○
28	Metaphor	Mountain View	○
29	Benham Capital Management	Palo Alto	×
30	Acer Group	Taiwan	○
31	Conner Peripherals Inc.	San Jose	○
32	Fujitsu Ltd.	Japan	○
33	Advanced Micro Devices Inc.	Sunnyvale	○
34	Digital	Maynard	△
35	Microsoft	Redmond	○
36	Xerox	Stamford	○
37	Hitachi Ltd.	Japan	○
38	National Semiconductor Corp.	Santa Clara	○
39	Advest Inc.	Hartford	×
40	Forrester Research Inc.	Cambridge	×
41	Apple Computer Inc.	Cupertino	○
42	Com Corp.	Santa Clara	△
43	Yankee Group	Boston	○
44	Robinson-Humphrey Co.	Atlanta	×
45	Siemens	West Germany	○
46	Dain Bosworth Inc.	Minneapolis	×
47	International Computers Ltd.	Britain	○
48	Mips Computer Systems Inc.	Sunnyvale	○
49	Carnegie Mellon University	Pittsburgh	×
50	Fujitsu	Japan	○

図 20 「IT 会社」対象の上位 50 個のレコード

Fig. 20 The top 50 records for "IT company" target.

ID	抽出パターン	抽出されたレコード数
1	ORG , LOC	1,031
2	ORG in LOC	828
3	ORG of LOC	390
4	LOC 's ORG	347
5	LOC , ORG	289
6	ORG , a LOC	248
7	ORG , based in LOC	122
8	In LOC , a ORG	17
9	ORG is based in LOC	14
10	In LOC , a spokeman for ORG	5

図 21 「石油会社」対象の抽出パターンおよびそのパターンで抽出されたレコード数

Fig. 21 Extraction patterns and the numbers of records extracted using the extraction patterns for "Petroleum company" target.

順位	会社名	場所	判定
1	Chevron Corp	San Francisco	○
2	Texaco	White Plains	○
3	Mobil Corp.	New York	○
4	Amoco Corp.	Chicago	○
5	Chevron	San Francisco	○
6	First Interstate Bancorp	Los Angeles	×
7	Imperial Chemical Industries PLC	Britain	×
8	Energy Security Analysis Inc.	Washington	○
9	Bechtel Group Inc.	San Francisco	×
10	Commodity Exchange	New York	×
11	AgResource Co.	Chicago	×
12	Exxon Corp.	New York	○
13	First Marathon Securities Ltd.	Toronto	×
14	Du Pont Co.	Wilmington	○
15	Electric Co.	San Francisco	△
16	Amoco	Chicago	○
17	Geldermann Inc.	New York	×
18	Texaco Inc.	White Plains	○
19	Whitney Leigh Corp.	Tulsa	○
20	Brooklyn Union Gas Co.	New York	○
21	Du Pont	Wilmington	○
22	Unocal Corp.	Los Angeles	○
23	Shell Oil Co.	Houston	○
24	Irving Trust Co.	New York	×
25	Coastal	Houston	○
26	Sparks Commodities Inc.	Memphis	×
27	Ashland Oil Inc.	Ashland	○
28	Project Inform	San Francisco	×
29	Mitsubishi Corp.	Japan	×
30	Associates Inc.	Tulsa	△

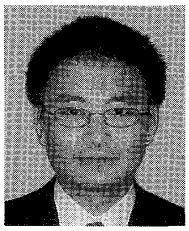
図 22 「石油会社」対象の上位 30 個のレコード

Fig. 22 The top 30 records for "Petroleum company" target.

ので，8 個はノイズがないものの，トピックに適合しないもので，39 個はノイズがなく着目するトピックのものである．石油会社を対象とした場合の上位 30 個のレコードについては，ノイズのあるレコード数，ノイズがないものの，トピックに適合しないレコード数および，ノイズがなく着目するトピックのレコード数はそれぞれ，2 個，11 個，17 個である．

(平成 19 年 3 月 20 日受付)
(平成 19 年 6 月 27 日採録)

(担当編集委員　波多野 賢治)



張 建偉（学生会員）

2005年筑波大学大学院システム情報工学研究科にて修士号取得．同年同大学院同研究科博士後期課程に移籍，現在に至る．ウェブマイニングと情報抽出に興味を持つ．日本データベース学会学生会員．

石川 佳治（正会員）

1989年筑波大学第三学群情報学類卒業．1994年同大学大学院博士課程工学研究科単位取得退学．同年奈良先端科学技術大学院大学助手．1999年筑波大学電子・情報工学系講師．2004年同助教授．2006年名古屋大学情報連携基盤センター教授．博士（工学）（筑波大学）．データベース，データ工学，情報検索等に興味を持つ．電子情報通信学会，日本データベース学会，人工知能学会，ACM，IEEE CS各会員．

北川 博之（フェロー）

1978年東京大学理学部物理学科卒業．1980年同大学大学院理学系研究科修士課程修了．日本電気（株）勤務の後，1988年筑波大学電子・情報工学系講師．同助教授を経て，現在，筑波大学大学院システム情報工学研究科教授，ならびに計算科学研究センター教授．理学博士（東京大学）．情報統合，ストリーム処理，データマイニング，情報検索等の研究に従事．著書『データベースシステム』（昭晃堂），“The Unnormalized Relational Data Model”（共著，Springer-Verlag）等．電子情報通信学会フェロー，日本データベース学会理事，ACM，IEEE-CS，日本ソフトウェア科学会各会員．