知能と複雑系　121－7
(2000. 7. 18)

関係学習を用いた対話的文書検索とWeb検索エンジンへの応用

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

〒226-8502 神奈川県横浜市緑区長津田町 4259

045-924-5218　okabe@ymd.dis.titech.ac.jp

あらまし　本研究では，検索エンジンとユーザを仲介し，適合フィードバックを実現する対話的な検索処理システムを提案する．このシステムは検索エンジンに与える検索式を，ユーザのフィードバック情報から自動的に生成する．検索式を生成するためにシステムは，ユーザが評価を行ったページを訓練ページとして，ユーザが欲しているページとそうでないページを判別するルールを学習する．検索式は，この判別ルールの条件部を変換することにより得られる．判別ルールはキーワード，関係演算子，検索範囲といったWebページの特徴を生かした構成要素からなり，Separate-and-Conquer 戦略と情報利得を評価関数とする探索により学習される．

キーワード　情報検索，適合フィードバック，関係学習，Web検索エンジン

Interactive Document Retrieval with Relational Learning and it's application to the Web Search Engine

Masayuki Okabe　　Seiji Yamada

CISS, IGSSE, Tokyo Institute of Technology

4259, Nagatsuta, Midori-ku, Yokohama-shi,Kanagawa 226-8503, Japan

045-924-5218　okabe@ymd.dis.titech.ac.jp

Abstract　This paper describes an interactive information retrieval system which mediates between the user and the web search engine to realize relevance feedback. This system automatically generates a query for search engine by converting rules which are learned to distinguish between relevant pages and unrelevant ones user has already seen and judged during retrieval. The rules consist of keywords, operators, and region restriction to represent the feature of the web page. They are learned by the algorithm which adopts separate-and-conquer strategy and top down heuristic search using information gain.

key words　Information Retrieval, Relevance feedback, Relational Learning, Web Search Engine

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Web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，それらを統計的手法を用いてランキングしたものを結果として返す．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，検索要求を反映した適切な検索条件を与えることが重要となる．

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ため，初めから適切な検索条件を設定することは難しい．このような場合には，検索を何度か行う中で結果を吟味しながら適宜検索条件を修正していくことが必要となるが，一般のユーザにとって，あるページ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ができても，その理由を検索条件として提示することは困難で負担のかかる作業である．よって，ユーザが適合ページの判定さえすれば，検索条件を自動生成してくれるシステムが望まれる．

このような状況に対処するには，適合フィードバック[6, 7]によるアプローチが有効である．適合フィードバックは，文書検索の分野で提案された，検索式を自動修正するための枠組みで，検索途中に見つけた適合文書を使ってユーザの検索要求を自動的に推定しながら，徐々に他の適合文書を集めていくという対話的アプローチを実現する．適合フィードバックにおいては，フィードバック情報から何を得て，それをどのように活用するのかが重要となる．

我々はこれまで適合フィードバックを使った文書検索において，適合文書を判別するルールを単語間の関係を使って学習する方法を提案し，このルールで適合と判別された文書から優先的にユーザに提示することにより検索効率が上がることを，新聞記事検索を使った実験において確認している．この学習アルゴリズムは文書を絞り込むために有効な，キーワードと関係演算子の組み合わせを提示することができ，Webページの検索においても同様に有効なものとなる．また，Webページはタグ付けによる構造化がなされており，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．よって，この構造的情報を生かすことでより効率的な絞り込みが行えると考えられる[1, 2]．

以上より本研究では，検索エンジンとユーザを仲介し，関係学習による検索条件の自動生成機能をもつ適合フィードバックを実現する対話的な検索処理システムを提案し，このシステムを使った検索過程について説明する．また，システムの中心的な機能となるWebページの特徴を生かした判別ルールの生成方法について考察し，その学習アルゴリズムについて述べる．

2 検索エンジン

2.1 適用対象の検索エンジン

現在WWW上には様々なタイプの検索エンジンが存在する．特定分野の検索エンジンを除く汎用型の検索エンジンは，主にディレクトリ型とロボット型に分けることができる．ディレクトリ型はツリー上に細分化されたカテゴリに予め人間の手によってページが分類・整理されているので，カテゴリをたどることにより情報を探すことができ，的を絞りやすく検索しやすいという利点がある．しかし，検索結果として提示されるページ数が少ないため，目的とするページが見つけられないことも多い．一方ロボット型は，ロボットと呼ばれるWWW巡回プログラムを使って集めた膨大なページを検索対象としており，キーワードや演算子などを使って全文検索を行うことができる．ディレクトリ型に比べ圧倒的に貯蔵ページ数が多いため，多様な要求に対応できることができるものの，検索条件をうまく与えなければ無関係なページが大量に提示されてしまうという欠点がある．

ロボット型検索エンジンにおいて，無関係なページが検索される原因は，検索条件によってページをうまく絞り込めていないことであると考えられる．実際，多くのユーザはキーワードを数語与えるだけに留まっており，演算子などを使って色々組合わせを考えるといったことは精通したユーザ以外ほとんど行われていないというのが現状である．以上より，本研究で提案するシステムは，ロボット型検索エンジンに適用することで，その効果がより発揮されると考えられる．

2.2 検索条件の分類

検索時に設定できる検索条件は，検索エンジンごとに様々である．以下では，代表的な検索エンジン[8]において設定可能な検索条件を調べ，その目的ごとに分類した．

- 検索対象の属性：多くの検索エンジンにおいて，Webページが持つ様々な属性を使った指定ができる．
  - 言語指定：ページが記述されている言語を選択する．25言語から指定できるものもある．
  - 地域指定：トップレベルドメインを指定することにより，そのページが存在する国や地域を指定することができる．

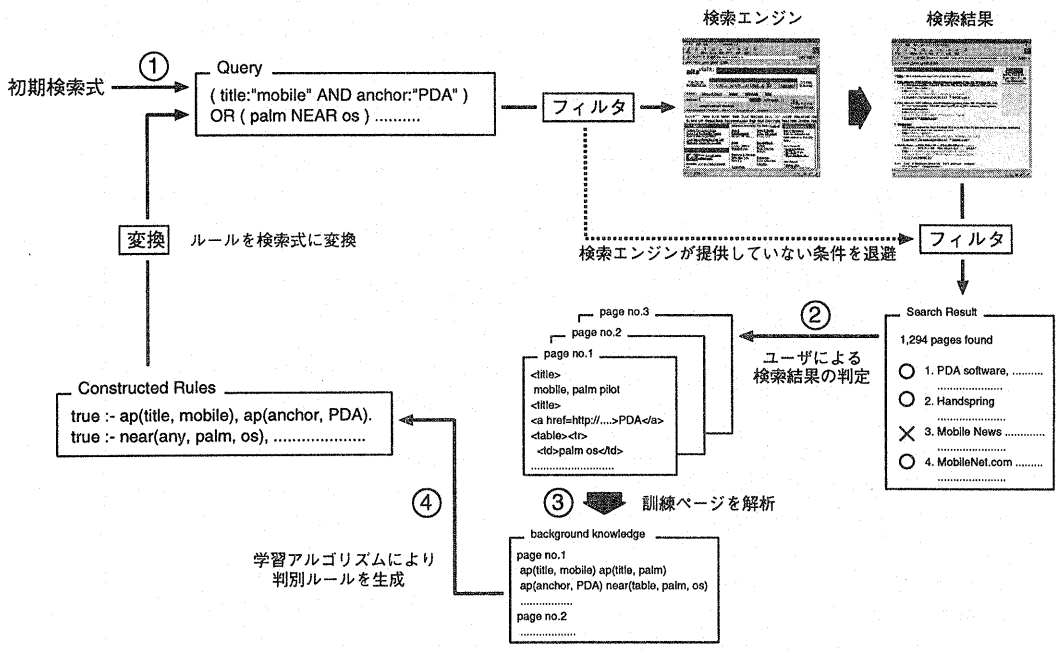

図 1: 対話的 Web 検索

- 日付指定：ページが最後に修正された日時を指定．その日に更新されたページのみを検索できるものもある．
- データタイプ指定：音声，画像，動画ファイルを含むページやニュースグループを検索対象として指定する．

- キーワードの属性：入力するキーワードの種類として，普通の単語，人名，フレーズといった指定ができる．また，自然言語文の入力を許すものもある．
- 演算子：複数のキーワードを入力した場合に，キーワード間の関係を指定できる．AND, OR, ANDNOT などの論理関係や NEAR といった近接関係などの演算子が使用できる．括弧を使った複雑な論理式を許すものもある．
- 検索範囲：ページ内の特定箇所，例えば，タイトル，アンカーテキスト，URL，画像ファイルの名前などにおいてキーワードが存在するかどうかを調べることができる．

これらの条件の中で，検索対象の属性とキーワードの属性はユーザによって予め与えられるべき条件であるため，自動的に獲得する検索条件の構成要素としない．よって本研究では，キーワード，演算子，検索範囲を用いた検索条件の自動獲得を行うこととする．このような検索条件は式として表せるので以下，検索式と呼ぶ．

3 適合フィードバックによる対話的 Web 検索

図 1 に，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を示す．このシステムの主な目的は，検索エンジンに適合フィードバック機能を追加することであり，ユーザが検索結果の一部に下した判定結果を使って，検索エンジンに与える検索式を自動的に生成する．このシステムには次のような利点があり，効率的な情報収集を行うための支援環境として役立つことが期待できる．

- ユーザは適合ページを判断するだけで，検索質問を練り直す作業を行わなくて良い．
- 検索結果の部分的評価を繰り返すことで，全結果を調べることなく効率的に適合ページを集めることができる．

以下，このシステムの全体の手続きを，各ステップで提供される機能と共に述べる．(各手続きは，図中の番号の付いた矢印における処理を指す)

1. **初期検索式**　ユーザは，まず少数のキーワードから構成される（通常簡単な）検索式を入力する．このシステムではその他，初期検索には用いないけれども検索に役に立つと思われる他のキーワードの入力も促し，後に生成する判別ルールの構成要素として役立てる．
2. **ユーザによる検索結果の判定**　検索エンジンに検索式を与え，その検索結果を得る．これをユーザに判定してもらい（通常上位 10 ページ程），情報が得られたページ（正例ページ）と情報が得られなかったページ（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**　訓練ページを解析する目的は，主に2つある．1つはキーワードの拡張である．キーワードの質と量が不十分であると，検索要求がうまく表現されず不十分な結果となってしまう．ユーザが最初に入力するキーワードは一般的にこの傾向が強いので，訓練ページ内で頻出する単語や既存のキーワードの関連語などを追加するなどして，キーワードを増やす作業が必要となる．もう一つは，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）や，他のキーワードと近接しているか等の情報を取得し，判別ルールを生成するための背景知識を作ることである．
4. **判別ルールの生成**　訓練ページを解析することにより得られた背景知識を使って，正例ページを含み負例ページを排除する判別ルールを学習アルゴリズムにより生成する．このルールの条件部は正例ページの特徴を表しているので，これを検索式に変換し，検索エンジンに与えることにより再度検索を行う．

検索は以上の手順で進み，4 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．フィードバックを行うかどうかは，そのときの検索結果により（例えば無関係なページが多く含まれている場合など）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．このほか，対象とする検索エンジンによっては次のような機能も必要となる．(図中の点線矢印での処理)

- **検索式のフィルタ**　用いる検索エンジンに関係なく作られた検索式を使いたい場合，検索エンジンで提供されてない条件を一度はずして検索エンジンに与えられる形にし，得られた検索結果から，退避させた条件が当てはまらないものを除外する機能．

4　判別ルールの生成

検索エンジンに与える検索式は，ユーザから提示された正例ページと負例ページを訓練データとする分類学習を行うことにより得られた判別ルールを変換したものである．ここでは，2 章で検討した検索式の構成要素を用いた判別ルールの表現と生成方法について述べる．この判別ルールの生成は，3 章の全体の手続きの 4 で用いられる．

4.1　判別ルールの表現

学習により獲得する判別ルールは，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．
- $not(region\_type, word)$: ページ内の $region\_type$ 部分に $word$ が現れない．
- $near(region\_type, word1, word2)$: 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．

検索エンジンで近接関係を指定できるものは現在ほとんどないが，この条件は文書検索を行う上で非常に有効なものとされている [4]．10 単語以内で順不同という条件は，検索エンジン AltaVista で提供されているものと同じ設定である．

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 ap, not, $near$ リテラルともに $region\_type$ でそのリテラルが成り立つ場所を指定している．$region\_type$ の種類は以下のものである．

- $title$: タイトルタグで囲まれたテキスト．
- $anchor$: リンクが張られているテキスト．
- img : IMG タグ内の ALT オプションで指定されたテキスト．
- any : ページ内の任意のテキスト．

これらのリテラルにより，例えば次のような判別ルール集合が生成される．

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

各ルールは代替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，またはページ内のどこかで"palm" と"os" が近接して現れているページを表している．

4.2 背景知識の生成

判別ルールを構築する前に，キーワードと *region_type* を具体的に組み込みことにより可能な全てのリテラルを用意し，それらが各訓練ページで成り立つかどうかを調べる．これを次の手順に従い行う．

1. 全てのキーワードに関して，それを引数とした *ap* リテラルと *not* リテラルを各 *region_type* ごとに生成する．
2. キーワード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* ごとに生成する．
3. 生成した各リテラルが各訓練ページ内において成り立つかどうかが記された真理値表 T を生成する．

例えば，キーワード集合が，

$$\{\ \text{mobile, PDA}\ \}$$

であった場合，全部で 24 個の以下のようなリテラル集合を用意し，これらのリテラルが示す関係が各訓練ページ内で成り立っているかどうかを調べ，真理値表 T としてまとめる．用意したリテラルは，判別ルールのボディ部（条件部）を構成するリテラルの候補であるから，これを**条件候補リテラル集合** C と呼ぶことにする．

$$\left\{\begin{array}{l} ap(title, \text{mobile})\quad ap(anchor, \text{mobile})\quad ap(img, \text{mobile}) \\ ap(any, \text{mobile})\quad ap(title, \text{PDA})\quad ap(anchor, \text{PDA}) \\ ap(img, \text{PDA})\quad ap(any, \text{PDA})\quad not(title, \text{mobile}) \\ not(anchor, \text{mobile})\quad not(img, \text{mobile})\quad not(any, \text{mobile}) \\ not(title, \text{PDA})\quad not(anchor, \text{PDA})\quad not(img, \text{PDA}) \\ not(any, \text{PDA})\quad near(title, \text{mobile, PDA}) \\ near(anchor, \text{mobile, PDA})\quad near(img, \text{mobile, PDA}) \\ near(any, \text{mobile, PDA}) \end{array}\right\}$$

入力：条件候補リテラル集合 C，真理値表 T
　　正例ページ集合 E^+，負例ページ集合 E^-．
出力：判別ルール集合 R．
変数：判別ルール *rule*，除外リテラル集合 S．

初期化
　$R \leftarrow empty$
　$S \leftarrow empty$
　$rule$:= $true$:-
Repeat
1: **If** *rule* が E^- のページを一つも満たさない **then**
2: 　· *rule* を R に加える．
3: 　· *rule* が成り立つページを E^+ から取り除く．
4: 　**If** E^+ が空集合となる **then** 終了
5: 　**Else** *rule* と S を初期化．
6: **Else**
7: 　· C 中の全てのリテラルにつき，情報利得 G を
8: 　　計算する．ただし，S 中のリテラルは除く．
9: 　**If** $G > 0$ となるリテラルがない **then**
10: 　　**If** *rule* にリテラルが一つも追加されてない
11: 　　**then** 終了
12: 　　**Else**
13: 　　· S を初期化し，*rule* に最初に追加された
14: 　　　リテラルを S に加える．
15: 　　· *rule* を初期化する．
16: 　**Else**
17: 　　· G が最大となるリテラルを *rule* と S に加える．

図 2: 判別ルール生成手続き

4.3 学習アルゴリズム

判別ルール集合 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 [3] を用いており，判別ルール（*rule*）を一つずつ生成し，R に追加する作業を繰り返す（図中の Repeat 以下 2 行目）．*rule* が一つ生成されると，それによって被覆されるページが正例ページ集合 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（3〜4 行目）．

また，*rule* は空のボディ部にリテラルを一つずつ追加していき（17 行目），負例を一つも含まなくなると完成となる（1 行目）．追加するリテラルは，条件候補リテラル集合の中から選ばれるが，その際の評価基準には，以下の式から計算される情報利得 G を用いる（7〜8 行目）．

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

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

$e^{\oplus}_{before}$, $e^{\ominus}_{before}$, $e^{\oplus}_{after}$, $e^{\ominus}_{after}$ はそれぞれ，リテラル追加前と追加後に被覆される正例ページと負例ページの数である．これにより，正例 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被覆するリテラルが選ばれ，追加される．

情報利得を用いた探索は，効率的である反面，山登り的であるため，探索が行き詰まることがある [5]．本手法では，このような場合にバックトラックを行う（9〜15 行目）．バックトラックが実際に行われるのは，負例をまだ含んでいるのに選択できるリテラルがなくなったとき，つまり，過去のいずれかの時点に戻ってリテラルの選択をやり直す必要が生じた場合である．この時，*rule* のボディ部が空の状態であると，バックトラックは行えないので手続きは終了となる（10〜11 行目）．バックトラックが行える状態であれば，最初に追加するリテラルの選択からやり直すものとし，できるだけ冗長な探索を行わないようにしている（13〜15 行目）．

5 まとめ

本研究では，検索エンジンに与える検索式を適合フィードバックを用いて自動的に生成するシステムの構成とその検索手続きについて説明した．このシステムは，ユーザより提示される訓練ページを判別するルールを学習し，得られたルールを検索式に変換する．我々はWebページの特徴を生かした判別ルールについて考察し，その具体的な学習方法についても提案した．

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いない．しかし，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．今後このシステムの実装を行うとともに，具体的な検索を行う中でこのシステムを評価していく予定である．

参考文献

[1] Chakrabarti, S. et al.: A New Approach to Topic-Specific Web Resource Discovery, In *Proceedings of WWW8*, pp.545-562 (1999)

[2] Marchiori, M.: The Quest for Correct Information on the Web: Hyper Search Engines, In *Proceedings of WWW6*, (1999)

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

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

[5] 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)

[6] Rocchio, J.J.: Relevance feedback in information retrieval, In *The SMART Retrieval System-Experiments in Automatic Document Processing*, Prentice Hall, Inc., pp.313-323 (1971)

[7] 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)

[8] goo: http://www.goo.ne.jp
ODIN: http://odin.ingrid.org
AltaVista: http://www.altavista.com
Google: http://www.google.com
Yahoo: http://www.yahoo.com
Excite: http://www.excite.com
Infoseek: http://infoseek.go.com
Lycos: http://www.lycos.com
findwhat: http://www.findwhat.com
WebCrawler: http://www.webcrawler.com
DirectHit: http://www.directhit.com
Thunderstone: http://www.thunderstone.com