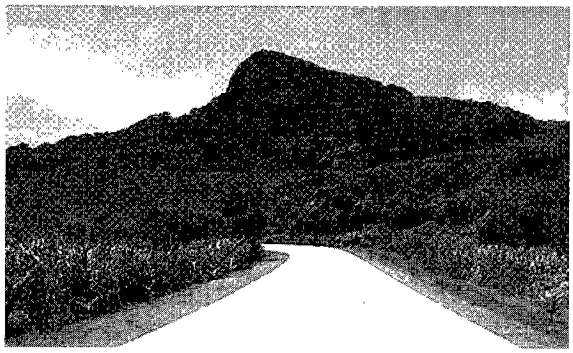
# 道しるべ：WWWサーチエンジンの作り方



原田 昌紀　harada@ingrid.core.ntt.co.jp
NTT未来ねっと研究所

## はじめに

サーチエンジンに代表される情報検索サービスは，WWWの普及と拡大とともにその重要性を増してきた．しかし，今日のサーチエンジンは網羅性，索引の鮮度，検索の精度，ユーザインタフェース，ネットワークの利用効率など多くの課題を抱えており，さらなる技術革新が必要である（**図-1**）．実際，使用していて不満を感じる場面も多いだろう．

本稿の目的は，インターネット情報検索研究を志す者に，サーチエンジンを用いて研究を進めるうえで参考になる情報を提供することである．どの分野にもいえることだが，実際に動作するシステムを用いて，現実のフィールドでアイディアの有効性を検証することは大切である．

なお，メタサーチエンジンのように既存の検索サービスを利用した研究も存在するが，利用できる機能が限定されるので本質的な改良ができないだけでなく，サーチエンジンの内部構造や改良などの情報が公開されないために実験結果の再現性が保証されないなど，研究として行うには不利な点が多い．

サーチエンジンの開発は技術的には難しくはない．必要なツールの多くはオープンソース・ソフトウェアにあるし，高速なCPUや大容量のハードディスクも容易に入手できる．ただし，運用面で注意を配らなくてはならないことは多い．

なお，Yahoo!☆1，Lycos☆2，Inktomi☆3，そして最近ではGoogle☆4に代表される多くの著名なサーチエンジンは，大学の研究に端を発している．高速なネットワーク接続など，大学の設備が自由に使える間がチャンスであり，特に本稿を読んでいる学生の皆さんの奮起を期待したい．

☆1 Yahoo!　http://www.yahoo.com/
☆2 Lycos　http://www.lycos.com/
☆3 Inktomi Corporation　http://www.inktomi.com/
☆4 Google　http://www.google.com/

## サーチエンジンを構成する要素技術

### ○WWWロボット

サーチエンジンを作るための材料として，まず検索対象となる大量のHTMLファイルが必要である．HTMLファイルを収集するための専用ソフトは，一般にロボットと呼ばれる．ロボットは（1）既知のURLのリストからHTMLファイルが未取得のURLを選択し，（2）HTTPによってWWWサーバからHTMLファイルを取得し，（3）そのHTMLファイルに含まれるリンク（URL）を既知のURLのリストに加える，といった動作を繰り返す．

HTTPにはサーバ上に公開されているファイルの一覧を得るための機能がないため，クライアントのみによる収集ではこのような非効率的な方法をとらざるを得ない．現実にはリンクで結ばれたページすべてを収集することは不可能であるため，（1）の収集対象のURLの選択が，サーチエンジンの性格と有用性に大きく影響する．

### ○全文検索エンジン

かつてのテキスト検索システムは，テキスト中から索引語を抽出する自動索引付け方式を採用するものが多かったが，1990年代以降はテキスト中に出現するほぼすべての語句を検索可能にする全文検索方式が主流になってきた．全文検索方式のメリットは，検索もれが生じにくいことと，索引語を統制するための辞書のメンテナンスのコストが低い点にある[1]．サーチエンジンにおいても，最初期にはタイトルや見出しだけを検索するシステムがあったが，現在では全文検索エンジンを用いることが一般的である．

全文検索エンジンは事前にテキストを索引付けし，転置ファイルを作成することで，高速な文字列検索を実現

する．索引作成時には，形態素解析を用いて，テキストを索引語に分割し，それらの生起位置を転置ファイルに保存する．検索時には，検索質問に含まれる検索語の出現位置のリストを転置ファイルから読み出し，その生起頻度などから質問とテキストの適合度を計算する．

サーチエンジンの場合には，HTMLファイルのエンコーディングおよび使用言語を判別し，言語に応じた形態素解析アルゴリズムを適用する必要がある．また，HTMLが持つタイトルや見出しといった構造を，語句の位置情報として格納し，検索時に利用するものが多い．

全文検索の索引のデータ構造としては，転置ファイルの他に，文字成分表，接尾辞配列（suffix array）などがあるが，サーチエンジンで用いられている全文検索エンジンの多くは転置ファイルを採用している．その理由として，テキスト自体にアクセスせずに転置ファイルのみで検索処理を完結できるため，スケーラビリティを確保しやすい点や，生起位置リストの圧縮により，記憶領域を節約できる点などが挙げられる．

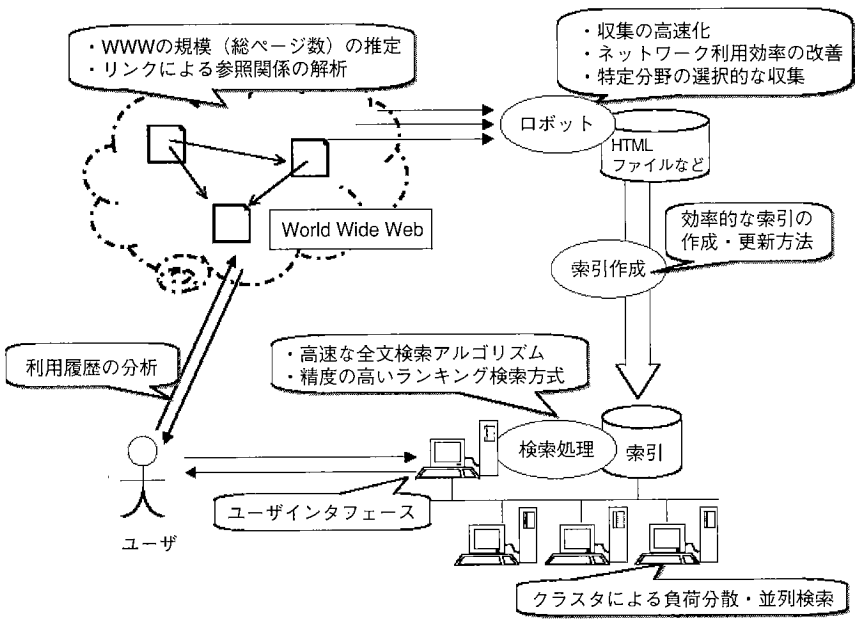



**図-1　サーチエンジンに関連する研究のポイント**

## 研究開発の動向

### ○ロボットの研究開発動向

単純な実装によるロボットはネットワークやWWWサーバへの負荷が大きく，収集に時間がかかる．そこで，大量のページを網羅する大規模なサーチエンジン，あるいは新鮮な情報を検索できるサーチエンジンを実現するために，並列・分散化を含めたロボットの高速化に力が注がれている[2]．山名らによる分散型WWWロボット実験は，ロボットを地理的に分散配置し，それらを協調動作させることで，高速かつネットワーク利用効率のよい収集を目指している☆5．

一方で，特定分野のページの選択的収集方法[3),4)]や，重要なページの優先的な収集方法[5)]なども研究されている．

### ○全文検索エンジンの開発動向

今日の世界最大級のサーチエンジンでは，約10億URLのページを検索対象とし，テラバイト級のテキストに対して，1秒間に数十回以上の検索質問を処理していると言われている．このような性能の要求から，メモリなどのリソース消費を節約した適合度計算方法[6)]や，クラスタによる並列検索[7)]などの大規模データベースの技術が重要となっている．

☆5 山名早人，分散型ロボット実験：http://www.etl.go.jp/~yamana/DWR/

### ○リンク解析を用いたランキング方式

サーチエンジン独自の展開を見せつつある技術として，ハイパーリンクの解析を利用したランキング検索方式が挙げられる．

情報検索システムの多くは，ユーザの質問と検索対象の適合度を計算し，適合度順に検索結果をランク付けして表示する．ランキング精度の向上は，情報検索分野の主要な研究課題であり，そのためにさまざまな検索モデルが考えられてきた[8)]．

しかし，かつての研究が想定していた検索対象が，新聞記事や特許明細書のような均質・静的なテキスト集合であったのに対して，WWW上のテキストはさまざまな言語・表現が用いられており，ひんぱんに更新される，非均質・動的なハイパーテキスト集合である．また，不特定多数のユーザに利用されるサーチエンジンにおいては，次のような状況もあることから，テキストと質問の類似尺度を用いた従来の適合度計算によるランキング検索では精度が不十分である．

- ほとんどの検索質問は1～2語であり，適合度を計算するための情報に乏しい．
- 検索対象ページ数が大量であるにもかかわらず，一度に表示可能な検索結果数は限られている．
- ランキングの上位に位置することを意図した行為（スパム）が存在する．

こうした状況において，WWWを有向グラフとしてモデル化し，ハイパーリンクの参照関係から，多くの人に価値を認められている有用なページを見つけ出し，ランキングに反映させる手法が注目されている．以下では代表的な手法として，Googleで用いられているPageRank[2),9)]とKleinbergによって提案されたHITS[10)]を挙げる．

**• PageRank**

PageRankは多くの価値のあるページからリンクされているページは，価値のあるページであるというように再

帰的に定義された重要度である（**図-2**）．従来はページの重要度として，そのページの被リンク数を用いることが一般的であったが，PageRankでは機械的に生成されたリンクの影響を受けにくいなどの利点がある．PageRankはランダムに動くユーザが，それぞれのページを訪問する定常状態確率とみなすこともできる．

**• HITS**

HITSは，ある特定のトピックに関するWWWグラフから，オーソリティとハブを抽出する．オーソリティとはそれ自身が情報を提供しているページであり，ハブとは多くのオーソリティをリンクするリンク集的なページである（**図-3**）．オーソリティとハブは相互再帰的な関係にあり，あるページのオーソリティスコアは，そのページを参照するページのハブスコアの和であり，ハブスコアはそのページが参照するページのオーソリティスコアの和である．

PageRankは，事前にWWW全体での重要度を計算しておくために，検索質問とは無関係な値になる．そのため，Googleでは一般的な適合度の計算と，PageRankを組み合わせたランキングを行っている．

一方，HITSでは文字列検索の結果から得られる比較的小さなWWWグラフからオーソリティを求めることから，検索質問に関連した重要なページを求めることができる．しかし，HITSを用いても，しばしば検索質問とあまり適合しないページがランキング上位になることがある．これはトピックドリフト問題（topic drift problem）と呼ばれる[11]．この問題は，リンクの接続関係に加えて，アンカーテキスト（HTMLにおいて，<A>…</A>で囲まれているテキスト）やアンカー周辺のテキストを用いて内容の適合性を考慮に入れることで改善できる[12]．

リンク解析は，自動分類や関連ページのマイニングにも有効である．たとえば，共通のハブから参照されることの多いオーソリティ同士や，共通のオーソリティを多く参照するハブ同士を求めることで，強く関連したページを発見する方法が提案されている[13]．

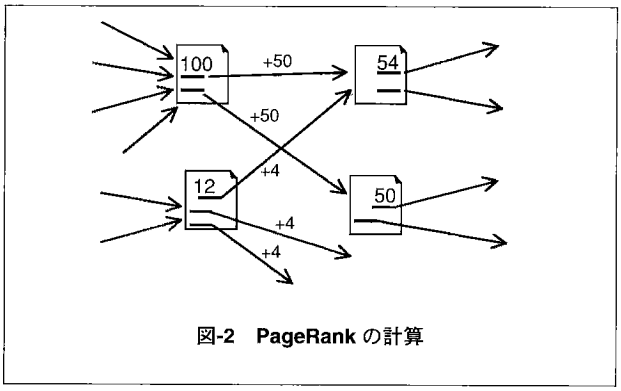



**図-2　PageRank の計算**

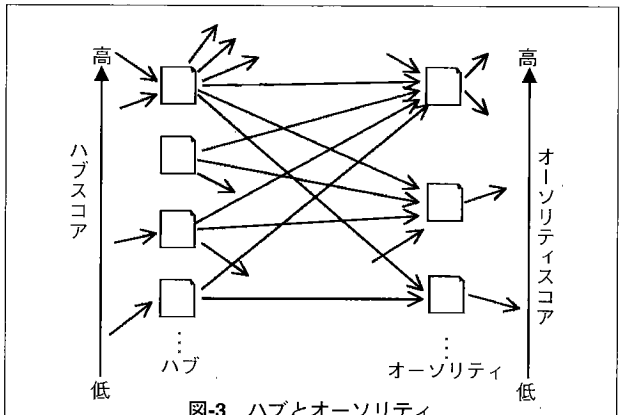



**図-3　ハブとオーソリティ**

## ソフトウェア・情報源の案内

### ○フリーの全文検索エンジン

ソースコードが入手可能な日本語全文検索エンジンは徐々に充実してきている．また，ChaSenなど優れた形態素解析ソフトウェアを利用すれば，日本語テキストを語句単位で索引付けするシステムの実装も難しくはない．馬場氏によるリスト☆6は多くの日本語全文検索エンジンを簡潔に紹介している．

ただし，フリーソフトウェアの日本語全文検索エンジンは，ランキングの精度などは重視されていない傾向にある．精度の厳密な評価が必要な場合には，適合度計算アルゴリズムなどを自ら実装するべきであろう．

高性能な全文検索エンジンの実装に興味があるならば，文献14）は必読である．全文検索の種々のアルゴリズムについて，特に性能面から詳細に述べている．ただし，マルチバイト文字や分かち書きなどの問題に関しては，日中韓の論文などにあたる必要がある[15)～19)]．

全文検索の入門的な解説は，文献1）や全文検索システム協議会☆7の活動報告にある．

### ○フリーのWWWロボット

先述したようにロボットはネットワークやWWWサーバに大きな負荷をかけるために，必要に迫られない限りは，ロボットを運用するべきではない．

単に大量のHTMLファイルでテストをしたいのならば，情報検索システムの評価会議であるTREC☆8で採用されたサーチエンジン評価用のテストコレクションの利用を検討してみるとよい☆9．テストコレクションによるサーチエンジンの評価とその課題については文献20）にて解説されている．

残念ながら，現時点では日本語のテストコレクションは存在していない．先述の分散ロボット同様，サーチエンジン研究のためのリソースの共有化を議論していく必要があるだろう．

これで不十分ならばロボットを自分で運用するしかないが，まずは大学内など限定した範囲での実験を行って問題を洗い出してから，徐々に範囲を広げるべきであ

☆6 馬場 肇，日本語全文検索エンジンソフトウェアのリスト：
http://www.kusastro.kyoto-u.ac.jp/~baba/wais/other-system.html
☆7 全文検索システム協議会：
http://www.asahi-net.or.jp/~zc7t-urb/
☆8 Text REtrieval Conference（TREC）：
http://trec.nist.gov/
☆9 Very Large Collection No. 2（VLC2）：
http://pastime.anu.edu.au/TAR/vlc2.html

る．また，自らロボットを実装する場合は，最低限のマナーとして，ロボット排除規約を遵守しなくてはならない☆10．これはrobots.txtというファイルで指定されたURLは収集しないという規約である．この他にも，ファイルの取得数や間隔の制御など，サーバやネットワークの負荷を低減するさまざまな配慮を行うべきである．

以下ではソースコードが入手可能なロボットの実装をいくつか挙げる．これらはいずれもロボット排除規約に準拠している．

- libwww-perl <URL:http://www.linpro.no/lwp/>
  libwww-perlはPerl用のHTTPやHTML関連のライブラリ集であり，さまざまなロボットの実装に使われている．Perlによる実装は大量のページを収集するのには不向きだが，多くの場合に必要十分な性能が得られると思われる．
- Iron33 <URL:http://verno.ueda.info.waseda.ac.jp/iron33/>
  早稲田大学の上田研究室で開発されているサーチエンジンVerno用のロボットであり，日本語のページの収集に適した実装となっている．
- W3C Webbot <URL:http://www.w3.org/Robot/>
  W3C（World Wide Web Consortium）によるHTTPやHTMLのリファレンスライブラリであるlibwwwの利用例であり，HTTPのパフォーマンスの計測などにも使われている．
- GNU Wget <URL:http://www.gnu.org/software/wget/wget.html>
  WgetはHTTPもしくはFTPでファイルを取得するためのコマンドである．再帰的にリンクをたどる機能があり，ロボット排除規約をサポートするので，簡易ロボットとして使うことができる．

## ○ディレクトリデータ

ロボット型のサーチエンジンよりも，ディレクトリ型の検索サービスに興味があるという場合には，Open Directory☆11の利用が考えられる．Open Directoryは，多くのボランティアがエディタとなって構築しているディレクトリ型の検索サービスである．このデータはRDF形式のXML文書として無償公開されており，一部を改変して利用することも認められている．エディタとして参加するのもよいだろう．

☆10 Koster, M., Robots Exclusion：
http://info.webcrawler.com/mak/projects/robots/exclusion.html
☆11 Open Directory：
http://dmoz.org/
☆12 Special Interest Group on Information Retrieval：
http://www.sigir.org/
☆13 International World Wide Web Conference Committee：
http://www.iw3c2.org/
☆14 the 6th RIAO Conference：
http://host.limsi.fr/RIAO/
☆15 VLDB Endowment：
http://www.vldb.org/
☆16 Search Engine Watch：
http://www.searchenginewatch.com/
☆17 ResearchIndex: The NECI Scientific Literature Digital Library：
http://www.researchindex.com/
☆18 Cora Research Paper Search：
http://cora.whizbang.com/

## ○文献の調査方法

情報検索分野では，ACM SIGIR☆12の国際会議が権威あるものだが，サーチエンジン関連の研究は，IW3C2☆13主催の国際会議WWWシリーズで発表されることが多い．他にマルチメディアよりの情報検索についての国際会議RIAO☆14，大規模データベースの国際会議VLDB☆15などにも発表がある．国内ではデータベース分野の研究会やシンポジウムを中心に発表が行われている．

D. Sullivan氏によるサイト☆16はサーチエンジンの規模などの情報がよくまとまっている．一般のユーザや，サイト作成者を対象とした記事が中心だが，論文や調査等が紹介されることもある．文献21）は情報検索分野の最近の研究を網羅しており，多数の文献を参照している．

なお，情報検索に限らず，計算機科学・工学の文献を調査する際にはResearchIndex☆17を利用してみることをお勧めする．ここで用いられているCiteSeer[22]というシステムは，ロボットを用いて米国の大学のサイト上に公開されているPostScriptないしはPDF形式の文書を収集し，それらを対象としてさまざまな検索機能を提供している．全文検索だけでなく，著者名などを抽出したうえで，引用関係によってオーソリティスコアの計算を行うなど，それ自体が先進的なサーチエンジンの実現例として参考になる．同様のサービスとして，Cora☆18も挙げられる．


参考文献

1) 学術情報センター編: 全文検索－技術と応用, 丸善 (1998).
2) Brin, S. and Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proc. of 7th Int'l WWW Conference (1998).
3) Chakrabarti, S., Dom, B. and van den Berg, M.: Focused Crawling: A New Approach for Topic-Specific Resource Discovery, Proc. of 8th Int'l WWW Conference (1999).
4) 横路, 高橋, 三浦, 島: 位置指向の情報の収集, 構造化および検索手法, 情報処理学会論文誌, Vol.41, No.7 (July 2000).
5) Cho, J., Garcia-Molina, H. and Page, L.: Efficient Crawling Through URL Ordering, Proc. of 8th Int'l WWW Conference (1999).
6) Cutting, D. R. and Pederson, J. O.: Space Optimization for Total Ranking, Proc. of 5th RIAO Conference, pp.401-412 (1997).
7) 沼尻, 竹岡, 渡辺, 芦川, 上田: 全文検索システムVernoのアーキテクチャの設計, 第2回インターネットテクノロジーワークショップ (WIT'99) (1999).
8) 徳永: 言語と計算5 情報検索と言語処理, 東京大学出版会 (1999).
9) Page, L.: PageRank: Bringing Roder to the Web, Stanford Digital Libraries Working Paper 1997-0072. <URL:http://www-db.stanford.edu/~backrub/pageranksub.ps>
10) Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment, Proc. of ACM-SIAM Symposium on Discrete Algorithms (1998).
11) Bharat, K. and Henzinger, M.: Improved Algorithms for Topic Distillation in a Hyperlinked Environment, Proc. 21st Int'l ACM SIGIR Conference (1998).
12) 風間, 原田, 佐藤: ハイパーリンクとアンカーテキストを利用した情報検索とランキングの一手法, 情報処理学会研究報告 SIGDD, Vol.24 (2000).
13) Dean, J. and Henzinger, M.: Finding Related Pages in the World Wide Web, Proc.of 8th Int'l WWW Conference (1999).
14) Witten, I. H., Moffat, A. and Bell, T. C.: Managing Gigabytes: Compressing and Indexing Documents and Images, Morgan Kaufmann Publishers (1999).
15) 赤峯, 福島: 高速全文検索のためのフレキシブル文字列インバージョン法, アドバンストデータベースシンポジウム'96, pp.35-42 (1996).
16) 松井, 難波, 井形: 高速テキスト検索エンジン, 情報処理学会研究報告 SIGDD, Vol.7 (1997).
17) Ogawa, Y. and Matsuda, T.: Overlapping Statistical Word Indexing: A New Indexing Method for Japanese Text, Proc. of 20th Int'l ACM SIGIR Conference, pp.226-234 (1997).
18) Nie, J. Y., Brisebois, M. and Ren, X.: On Chinese Text Retrieval, Proc. of 19th Int'l ACM SIGIR Conference, pp.225-233 (1996).
19) Lee, J. H. and Ahn, J. S.: Using n-Grams for Korean Text Retrieval, Proc. of 19th Int'l ACM SIGIR Conference, pp.216-224 (1996).
20) 福島: WWW情報検索技術と評価の問題, 情報処理, Vol.41, No.8, pp.913-916 (Aug. 2000).
21) Baeza-Yates, R. and Ribero-Neto, R.: Modern Information Retrieval, Addison-Wesley (1999).
22) Giles, C. L., Bollacker, K. and Lawrence, S.: CiteSeer: An Automatic Citation Indexing System, Proc. of 3rd ACM Conference on Digital Libraries, pp.89-98 (1998).


（平成12年10月12日受付）