修 士 論 文

WWW 上のリンク構造を用いた情報検索に関する研究

指導教官　篠田 陽一 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

大島 龍之介

平成 12 年 2 月 15 日

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1 背景と目的

World Wide Web(以下 WWW、Web と略する) などのハイパーテキスト、ハイパーメディアはリンクによって、非シーケンシャルな構造を持っている。そもそもハイパーテキスト、ハイパーメディアは大規模な情報を利用者が把握・利用できる手法として、また、各種の主要なメディアを統合的に扱える手法として、使い易い利用者インタフェース、大規模な情報の共有、検索、グループウェアの様な思考支援などに利用されてきた。

コンピュータやネットワークの飛躍的な進歩で、大規模な情報が扱えるようになるにつれて、ハイパーメディアの重要性は増していくことになる。Web は、ハイパーメディアの歴史の中で最もインパクトを与えた技術である。今日、ハイパーメディアで利用する技術の中でも最も普及しているというだけでなく、世界中から高速に情報を入手でき、また個人レベルでの情報の発信・共有が可能で、時々秒々と追加、更新されていく。情報産業を支える、まさにインフラストラクチャとなる技術として成功しているものといえよう。

ハイパーテキスト、ハイパーメディアは従来の文章にはない柔軟度を持っている。しかし、シーケンシャルでないがゆえに、利用者が戸惑ったり、迷ったりする恐れもでてくる。特に膨大な情報の中では、利用者が迷わずに適切な情報にたどり着けるように全体を俯瞰してうまく誘導する必要が出てくる。Web はまさに膨大であるが、そもそも Web はこの全体を俯瞰して、ユーザを誘導する機能に乏しい。

Web には、全体を俯瞰してユーザを誘導する機能が備わってはじめて、利用者自身が情報全体を正確に把握することができるようになるといえる。そして、そのためには Web 上の情報の自動的なグループ化、階層化、順序づけをする機能が必要となってきた。

上記の問題を解決するために、従来より Web 上の情報の言語学的分類が行なわれてき

た。その結果、キーワードから Web の情報を整理して利用者に提供をし、誘導をするシステムが作られた。検索エンジンと呼ばれるこれらのシステムは、利用者に広く用いられるようになった。

しかし、既存の検索エンジンは Web の情報をグループとして効率よく取り扱う機能にいまだに欠けている。Web の特長であるリンクを分類の手がかりとして利用することで、より精度が高く効率的な情報の整理が可能となる。例えば、Web のリンク情報の性質をうまく取り扱うことにより、Web ページ群の中でグループ化、階層化、順序付けすることができる。2 つの Web ページ間の距離 (リンクをたどる回数) が関連の度合を測る指標として機能すると考えられる。また、リンクによる Web 情報の分類は、単語をキーとする分類などの他の分類方法と組み合わせることにより、より細かく Web 情報を分類するための助けとなる。Web ページ制作者のリンク生成の意図を推定できれば、Web 利用者をよりうまく誘導できるようになるだろう。リンクより構造の情報を抽出できれば、Web 情報を分類で得られた構造の情報を元に、以下のような分野での応用が期待される。

- Web 情報の検索エンジン
- Web 情報のダイジェスティング
- Web 情報のプリフェッチ、キャッシング
- Web 情報の別の視点からの再構成

本研究では、リンク構造の情報の性質を検証、分類して、それらを Web 上の情報より抽出する方法を考案する。そして、抽出した情報をもとに Web 上の情報をグループ化、階層化、順序づけする方法を提案する。

さらに提案した方法の実際の応用例として、様々な情報検索、および従来の情報検索の改善をおこなう。

提案の内の一部の部分は実験のためのプロトタイプとして実装をおこない、その有効性について調べるための実験をおこなった。

1.2 本論文の構成

本論文は、8 つの章より構成される。

第 2 章では、本研究に関連する Web 上の情報検索の背景と研究を詳しく述べる。続いて、第 3 章で本研究の対象とする Web、Web のリンク構造、Web における情報検索に対

するモデルを定義する。この章のモデルに立脚して、以降の論を進める。第 4 章より本論に入るが、まず Web のリンク構造情報の性質を詳細に検証、分類し、それらの性質を抽出する方法を提案する。第 5 章では、リンク構造情報を応用して、新しい情報検索、あるいは既存の情報検索を改善する様々な方法を提案、評価する。第 6 章では、第 5 章までに提案した方法の一部を実装したプロトタイプについて説明した後、プロトタイプで行った実験の説明、および、その結果の考察を行う。第 7 章では、本研究では達成できなかった部分に関して検討し、さらに本研究を発展させたときの長期的な展望を示す。最後に第 8 章で本研究をまとめて、本論文の締め括りをする。

第 2 章

Web 上の情報検索

Web は 1994 年ごろより、Internet 上の重要な応用技術として、急速に普及をした。しかし逆に、現在にまで渡ってあまりにも急速に普及し、その全体の情報量は増大し続けているため、多くの問題を新たに引き起こしてきた。

その 1 つが Web 上での情報検索の問題である。Web には多数の情報全体のディレクトリ、インデックスを扱う能力が備わっていなかった。また、Web では情報の利用のみならず、情報の発信の敷居が低い。そのために、質が様々で雑多な情報が時々刻々と追加、更新されている。このために、あまりに情報過多な環境となってしまい、利用者と発信者の双方どちらにとっても、意図に沿った Web の利用が困難になっている。

この問題を解決するために、検索エンジンと呼ばれる、Web 上の情報の収集、及び検索を可能とする技術が開発され、利用者が適切な情報を高速に取り出すための Web 上のサービスとして、現在でも主要な地位を占めている。現在の検索エンジンは、主にロボット型とディレクトリ型の 2 種類に分類することができる。

2.1 ロボット型検索サービス

ロボット型の情報検索システムでは、検索システムがロボットと呼ばれるアプリケーションにより Web 上のデータを事前に自動的に収集する。そこから、Web 上の各種情報から収集・抽出された単語を検索キーとしてインデックスを作成しておき、利用者はそのインデックスを利用して高速な情報検索をする。

ロボット型検索エンジンによる検索は、本の中の目的の箇所を取り出すために、キーとなる単語の索引を引き、それらを組み合わせているものであるといえる。

検索のキーとなる単語は自動的に索引化されており、高い再現率を達成できる一方で、

その適合率はかなり低くならざるを得ない問題がある。

例えば、内容には関係のない検索キーがたまたま出現した情報が多数あると、その中に適切な情報が埋もれてしまう。そして、1 つの言葉で多数の意味を持つ多義語はさらに対応が難しい。例えば、「パイソン」での検索結果は、実際の蛇である「にしきへび」、イギリスのお笑いグループ「モンティパイソン」、銃の「コルトパイソン」、オブジェクト指向スクリプト言語である「パイソン」、などが混在した結果となる。

また、内容が適切でも検索キーワードが出現しない情報はヒットしないので、これだけでは同じ意味を違う言葉で表現可能な同義語が存在する場合、結果が統合されずに分かれて出てくることが挙げられる。例えば、従来の全文検索では「サーチエンジン」と「検索エンジン」では、異なる結果が帰ってくる。

これらの問題の解決のために、検索システムは検索語の文書内の位置などの言語学的に発見されている様々な手法を採り入れている。しかし、そもそもグラフィック中心などで単語が少ない Web ページや、Web ページ群で 1 つのコンテンツグループの場合には対応できない。例えば「北陸先端科学技術大学院大学」で検索しても、http://www.jaist.ac.jp/index-jp.html は結果には (単語が存在しないので当然ながら) 返されない。

2.2 ディレクトリ型検索サービス

Yahoo![1] を代表とする従来のディレクトリ型の情報検索では、高い適合率を達成するために人為的にインデックスを作成する。索引のようなロボット型に対して、ディレクトリ型検索エンジンによる検索は、本の中の目的の箇所を取り出すために作られた目次から、キーとなる単語を取り出すようなものといえる。

前節のロボット型検索サービスで挙げた様々な問題は、人が意味を考えて分類することにより解決されて、利用者は無駄、冗長な情報を避ける事ができ、適合率は高くなる。しかし、人手を介することには、さまざまなコストがかかるため、検索システムが扱うことのできる Web ページの総量が少なくなり、再現率が低くなってしまう。また、インデックス作成の時間的なコストもロボット型と比較して高く、Web 上の情報の変化が検索へ反映されるまでに時間がかかる。そのために、なるべく多くの部分が自動化されることが望まれている。

2.3 最近の動向

ロボット型検索、ディレクトリ型検索に加えて、これらを複合して利用するメタ検索も存在している。また、個人のページ向け [5] の検索や、書籍情報や商品情報を提供して、情報の質を高めている場合や、特定のカテゴリに特化した検索も増加している。さらにWeb ページのだけでなく、ニュースなどの他媒体の情報を扱うものも増えつつある。

多くのポータルサイトでは検索機能が付加され、必須の機能の 1 つになりつつある。日本では Namazu[28][31] などを利用した検索機能の追加が良く見受けられる。また、企業が専門に扱っているデータを検索エンジン側に提供した形の検索も増加していて、検索はWeb の仕組みの中に着々と幅広い形で拡散、浸透しつつある。

ロボット型、ディレクトリ型いずれのシステムにせよ、より適合率の高い、自動化されたシステムが求められており。近年では、ロボット型とディレクトリ型の両方の機能を合わせ持たせる方向に進んできている。

2.4 最近の研究

従来の情報検索に対する評価では、文章全体に対する適合率と再現率が評価の基準であった。しかし、何千何万という Web ページが一度の検索で返されても、利用者が実際にの利用する範囲はせいぜい数十個の Web ページぐらいである。つまり、どのようにして上位となる Web ページの順序を決定するかも情報検索の大きな問題である。したがって、以下のような点も情報検索の重要な評価基準となってきている。[25]

- 利用者が取り扱える範囲内で見た場合の、検索結果の Web ページの適合率
- 利用者が取り扱える範囲内で見た場合の、検索結果に含まれる冗長な Web ページの割合
- 利用者が取り扱える範囲内で、効率的な絞り込みができるかどうか？

ここで、従来の順序付けを行う検索システムでは、その結果の順序付けが重要になってくるのだが、従来のロボット型検索エンジン AltaVista[2][32] では、1997 年時点では、以下のような方法で、検索対象のページに得点を付けている。

1. 単語の珍しさ

- 単語の珍しさを tf・idf 法[1]などの方法により得点にする
- 極端に一般的な語は無視

2. 検索語の文書内の位置

- タイトル内か、先頭近くの単語の得点を増す
- 目的の単語間の距離が近いと得点を増す
- 単語の出現回数 (2 回まで) が多いと得点を増す

この時期以後の AltaVista では、さらにページのダイジェストとしてそのページの一部を切り出している。また、リンクされている、リンクしているページの検出が可能となっている。

このようにキーワード検索では、順序の付け方は主に tf・idf 法などの単語の珍しさを基準にしているが、内容には関係のない検索キーがたまたま出現した情報に弱い。また、商業的なサイトが検索システムに高い順序を付けてもらおうとして、あまり内容に関係のないキーワードをわざと Web ページに追加する (スパミング) という問題もある。

また、絞り込み検索の観点から見た場合、キーワード検索で利用者が入力に使うキーワードの数のはせいぜい 2 から 3 である。2、3 のキーワードのみでは結果にまだ多数の Web ページが残り、十分な検索の絞り込みを行うのに不十分である場合が多い。しかし、利用者がさらにキーワードを着想することは 1、2 個目よりもかなり難しくなり、今までの Web のテキストだけを対象としたキーワード検索は限界に至りつつある。

これらの問題の解決に、Web の大きな特長であるリンクを使う研究がいくつか始められている。

情報をカテゴリに自動割り当てする研究 [12] では、Web ページのディレクトリ型の分野わけを、リンクを使って自動的に行なうものである。テキストのみの場合と比較して、リンクを使うことにより、適合率、再現率が共に増加することが示されている。

また、利用者の視点をリンクのつながりの周囲に拡大できるようにする研究として、Web のリンク構造を利用者に分かりやすいように可視化して、3 次元空間で注目する Web ページ周辺を取り出す納豆ビュー [24] や魚眼でとらえる研究 [23] などが挙げられる。

また、Cha-Cha[26] のようにイントラネット内部のリンクを視覚的に表示して、ユーザの誘導を手助けするシステムも登場している。

リンク構造によって、キーワード検索を改善する新しい研究、リンク構造を用いた解決例では、google[3] やクレバープロジェクト [22] などが挙げられる。例えば Google.com で

[1] Term Frequency, Inverse Document Frequency 法

は、良いサイトの基準は再帰的に多くのリンクの流入があることであるとして、リンク (されている or している) ページの得点を増加している。これを再帰的に繰り返すことにより、信用度ともいえるものをベースとして検索結果の順序を決定している。

リンクは意味的なつながりが大きいので、google では同義語の問題がかなり解決されている。

google ではさらに各単語の距離を考慮し (結果としてフレーズも検出して)、サイトの URL が一致した場合のグループ化を行なっている。

クレバープロジェクトでも、信頼度が高いページ「オーソリティー」と、それらへのリンクの集まりである「ハブ」を、リンクを使った再帰的な計算で求めている。

google やクレバープロジェクトのでは、逆に Web ページの分類がされていないので、Web 空間を分割して扱う機能が弱いところが問題である。また、信頼度が収束するまで再帰的に計算をする必要がある。

本研究では、このリンクの重要性を鑑みて、リンク構造を最大限に情報検索に利用することを目的としている。そのために、リンクの性質をより細かく分類して、分類を元に Web ページ群をグループ化、階層化、順序づけする方法を提案する。さらに提案した方法の実際の応用例として、様々な情報検索、および従来の情報検索の改善を行なっていく。

検索エンジンは企業によって開発されたものが多数を占めているため、これらの手法はブラックボックス化していて、アルゴリズムやパラメータなどの、情報が公開・共有されにくい点も、発展の障害の 1 つとなっている。

第 3 章

研究対象のモデル

本章では、問題をより具体的にとらえるために、Web、Web のリンク構造、Web 上の検索のモデル化を行う。

3.1 Web のモデル

本論文で使われている、用語の示すところを表 3.1で定義する。(図 3.1参照)

用語	定義
(Web) オブジェクト	Web サーバ上にある提供実体
(Web) リンク	Web オブジェクト間の参照
(Web) ページ	1 つの HTML ファイルを中心にした Web オブジェクトの集合による表現
(Web) コンテンツ	リンクされた Web ページの集合の中の意味的まとまり
(Web) コンテンツグループ	Web コンテンツの中で特に複数の Web ページが集まっているもの
Web 空間	Web コンテンツ全体の集合

表 3.1: 用語の定義

Web オブジェクトは、各 HTML ファイル、イメージファイル、サウンドファイルなど、Web サーバ上で提供されるファイル、あるいはファイル相当のもの (サーバが返すディレ

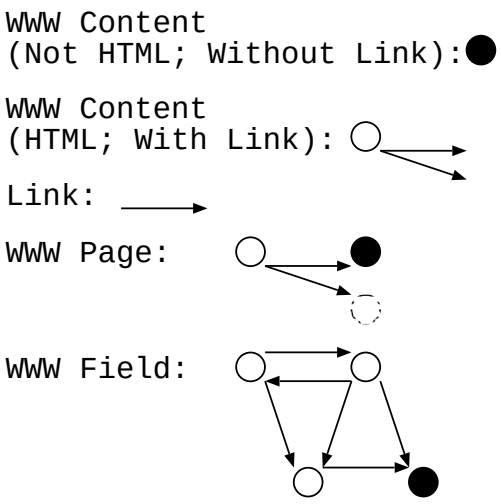

図 3.1: Web のモデル

クトリ情報など) である。URI[1][17] によって一意に特定される。サーバから TCP[2]/IP[3] 上の Hypertext Transfer Protocol[19](以下 HTTP と略) を介して、利用者にオブジェクトが提供される。

Web オブジェクトの参照は、Hypertext Markup Language[18] (以下 HTML と略) のリンクによって表現される。初期の HTML では、リンクは最初は一定の文字または画像からなるエリア (アンカー) が、他のメディア (の先頭または一部のエントリポイント) への参照を示しており、これを解釈するブラウザが、リンクより新しい情報を順次取得するものであり、現在でも広く利用されている。その後、Web のリンク構造を表現する手段は、単純なアンカーにとどまらずにフレームセットなどが導入されて発展を続けている。HTML4.0[13] や XML1.0[14] ではリンクのタイプを指示したり、指示するエリアの指定、双方向リンク、リンク群なども規定されるようになっている。本研究では、これらのもの全てをリンクとして扱っていく。

Web ページは 1 つの HTML ファイルを中心に、イメージファイルやサウンドファイルなどがリンクで組み込まれている表現である。ブラウザで 1 度に表示されている部分が大体これにあたる。本研究ではリンクの構造を扱うことから、Web ページの中でも特に、リンクを表現している HTML を中心に扱う。また、キーワード分類による検索システム

[1]Universal Resource Identifiers

[2]Transmission Control Protocol

[3]Internet Protocol

の改善が目標の 1 つであるので、リンクのなかでも、文章である Web ページ (HTML) へのリンクを中心に扱っている。

Web コンテンツは意味的にひとつのまとまりと見なされる部分で、複数のリンクされた Web ページ群で構成されることもあるし、Web ページの部分、部分で意味が分かれていれば、その部分部分が Web コンテンツである。Web コンテンツの中でも複数のページから構成される Web コンテンツの発見が、本研究の目標であるので、これらを特に Web コンテンツグループと区別して表記する。

Web 空間は検索の対象として扱う Web ページ、あるいは、Web コンテンツ全体の集合である。各 Web ページは、一般には他の Web ページへのリンク（HTML 上の表現では、他の Web オブジェクトへのリンク）を含む。多数の Web ページ同士が様々にリンクし合うことで、複雑にリンクされた Web 空間を構成している。

3.2 Web の利用者

続いて、本研究では、Web においての利用者の行動を、以下の 2 つに大別する。

1. 直接 URI を指定してページを取得する
2. ページのリンクをたどって周囲のページを取得する

別の言い方をすれば、利用者が次に閲覧するページの URI を得る方法が、現在の閲覧中のページから直接得る場合と、それ以外の場合に分けて考えるのである。

利用者はまず URI を直接指定して、目的のページ、あるいは目的に近いページを選びだす。その URI の取得方法は、検索エンジンでの検索による以外にも、ニュース・メール・口コミなどからであったり、履歴 (bookmark ファイル) で記録しておいたり、と多岐にわたる。

利用者は、直接 URI を指定してページを取得した後、続いて、そのページからリンクをたどって周囲のページの取得をいくつか繰り返す。多くの Web ページは他のページのリンクを含んでいる。すなわち、リンクをたどった先のページには新たなページへのリンクが含まれていることが多く、Web の利用者は、リンクを連続してたどって適切な情報を求めることになる。やがて、利用者はいくつかのリンクをたどって情報の収集を終える。

直接 URI を指定してページを取得する中でも、本研究のターゲットである検索をさらに詳しくモデル化すると、以下のように

1. キーワードの選択

2. 検索システムの結果が適切なものかを吟味

3. 結果のリンク先の内容を実際に確かめて適切か吟味

4. (目的の情報の獲得)

という手順を (場合によっては繰り返し) 行う。まず、目的の情報に関連するキーワードを選択して、検索する。得られた検索結果のタイトルやダイジェスト、さらには、実際にリンクをたどることなどから、目的の情報を探すことになる。目的の情報がうまく取り出せないときには、検索の絞り込みや拡大をするために新たなキーワードを選択して、再度検索を行うことになる。

検索の効率を上げるためには、各手順でなるべく利用者の手間を省く方式にする必要がある。(本研究では扱わないが、検索システムの処理速度自体の向上も必要である。) 利用者の手間を省くための具体的な目標は、Web 空間上で、情報検索の結果から利用者が人間的に手繰れる手数で目的とするコンテンツにたどりつけること、ユーザが人間的に手繰れる手数内のコンテンツの適合率を高めることである。本研究ではリンクに着目して、この検索の効率を改善するために「検索システムの結果が適切なものかを吟味」、「結果のリンク先の内容を実際に確かめて適切か吟味」する部分を (人手に代えて) システム側でなるべく自動化して、利用者の補助をする。

「検索システムの結果が適切なものかを吟味」する効率を良くするためには、利用者が把握できる範囲で提供される、グループ単位での情報の適合率を高めることである。現在の検索結果では、複数の Web ページで構成された情報は、ばらばらに提示されるために冗長であり、1 つのページにまとまっている場合よりも各ページの情報量は少ないので、検索結果の上位に入りにくい。そこで単に Web ページ単位で検索を行うだけでなく、リンクによって結果内の冗長な Web ページのグループを検出して、結果の縮退をして冗長度を下げる。リンク構造でつながった一連のページ群は、まとまった内容である可能性が高い。これらの複数の Web ページにまたがる情報は、検索結果として 1 つのまとめて、グループ単位で評価値を算出し、適切にダイジェストにして提示することにより、検索の効率が改善される。これらの方法により、利用者が手数以内の部分 (検索結果の上位部分) の適合率が高まり、より効率的な検索が可能となる。

このまとまったページ群をグループとして見つけ出す方法を、後の 5章、「リンク構造情報を用いたグループ化」で説明する。

「結果のリンク先の内容を実際に確かめて適切か吟味」する効率を改善するためには、検索結果のリンクをたどる回数を減らすことが必要である。検索の結果から実際の情報を調べる際には、いくつかのリンクをたどって全体を把握することが多い。また、複数

の Web ページで構成された情報が木構造 (あるいはそれに近い構造) をしているときに、葉のページのみを検索の結果として提示されると、全体の把握が難しくなる。つまり、グループ単位の Web ページ群では、より抽象度の高い、あるいは、先頭にくる Web ページの提供が望まれる。この結果内の吟味を効率的にするためには、リンク構造の情報を用いて、複数のリンクされた Web ページをグループ化する。続いてリンクの構造上、この Web ページグループを把握しやすいエントリポイント (例えば木構造の根) を、グループから検出して結果に提示する。

これらを実現する階層化、順序付けをする方法は、5章の「リンク構造情報を用いた階層化、順序付け」で説明する。

第 4 章

リンク情報の分類と抽出

本章では、Web 上のリンクの分類を行ない、この分類を元にリンク構造情報を抽出する。本章のリンクの性質に基づいて、その後の章で、リンク構造情報によるグループ化、階層化、順序づけが行われる。

4.1 Web のリンク構造

Web ではありとあらゆる文献を手軽に利用することができる。しかし、Web 自身には文献を整理、検索する機能は備わっていないので、それらはごちゃごちゃになっていて、利用者が必要とする文献を取り出すことが困難になっている。Web は一つの部屋のなかに百科事典から個人のメモや日記までが雑然と置かれているようなもので、いくらその中に必要な文献があるとわかっていてもそれでは利用者は困るのである。そこで検索エンジンと呼ばれる、Web の案内を行うサービスが発展してきた。しかし、既存の検索エンジンは、先ほどの例えでは、本のページのつながりを無視して、ページ毎に索引をつけるようなものである。そこで、本自身を検索の対象とするような新しい検索エンジンが必要となる。しかし本は一目で一まとまりだと認識ができるけれども、Web ではどうやってそのまとまりを見つけ出せば良いのであろうか。そのための重要な手がかりがリンクである。どのようにリンクが役立つか、ここで詳しくリンクの性質を調べていく。

リンクは文章の流れや参照を明示的に表現したものである。Web コンテンツ制作者が、リンク先の Web コンテンツとの間に、何らかの関連を認めてリンクを作成する。そこには一般的に人為的作業が関わっているので、キーワードには現れないような機械的には抽出しにくい関係でも、リンクによって取り扱いが容易になる可能性が高い。Web ページ制作者にとってリンクを作成する行為は、ある程度のコストがかかるので、リンクは (少

なくとも Web ページ制作者にとって) そのコストに見合う価値のあるものを意味している。Web の利用者にとっても、リンクをたどることはある程度のコストがかかるので、そのコストに見合う以上の価値をリンクをたどる先の情報に求めている。これらのリンクのコストを考えると、2 つの Web ページ間の関連の度合は、リンクの数やリンクで隔てられた距離 (リンクをたどる回数) に反映されていることが期待される。

さらに、Web 空間では個々の Web ページがリンクによって集団を形成し、それらが次第に大きな集団を形成するという、いわばボトムアップする形で形成されている「ハブ」の部分と、ある程度の権威を持った組織などによるトップダウン的な「オーソリティ」部分がが絡み合って構成されている。[22]「ハブ」と「オーソリティ」は必ずしも明確に分かれるものではないが、結果として、Web 空間は、関係が密 (リンクの数が多い、双方向に結び付いている) である Web ページ群が、他の密な Web ページ群と疎 (リンクの数が少ない、一方向に結び付いている) に結び付いている構成となっているものと仮定できる。Web 空間をリンク構造から眺めて、Web ページの間にリンクに基づいた関係 (例えば、直接リンクされているか否か、リンクを何回たどって到達できるかなど) をうまく定義することができれば、Web 空間を Web コンテンツへと自動的に分割する仕組みへの道が開けることになる。

利用者の利便性を考えた場合、Web コンテンツを見つけるリンク構造は検索システムの改善に大きく貢献する。リンク構造から、利用者に扱いやすいように、Web 空間を分割、縮退、あるいは可視化することができるようになる。リンク構造による縮退とは、リンク構造で Web 空間が密な部分をコンテンツグループとして、擬似的に 1 つのページとして扱うことである。適切な縮退により、各ページ個々には単語があまり含まれない場合にも、適当な検索の対象として選びだすことができる。

さて、Web ページの制作者は、他者の承認などの必要なく他の Web ページへ自由にリンクを作成することができるので、極めて種々雑多なリンクが Web 空間では存在している。ここでしかし、現在の使われているリンクは HTML の <a href=URI> タグによるリンクが大多数であるということが、問題の 1 つとして浮かび上がってくる。本でのページのならびから、他の参考文献を指し示すことまでのすべてがこの同じリンクで表現されているわけである。Web ページ制作者がリンクを形成する意図は実に様々であり、おおざっぱにいっても以下のようなものがみられる。

- 章間の構成を示す。
- 脚注など、詳細な情報を示す。
- よりメタな情報へのポインタ。

- 関連情報への参照。
- 誘導、順路を示すため。
- 用語集などへの参照。

これらの意図の違いから、リンクのページを結び付ける強さだけを考えても、当然のことながらその強さは異なってくる。例えば、文章の流れを示して誘導するリンクは、用語集などへのポインタとしてのリンクよりは、強くページを結び付けていると考えられる。また、メタな情報へのものか、それとも詳細な情報へのものかわかれば、Web 利用者をよりうまく誘導できるようになる。本研究の目的の 1 つは、このリンクの強さと役割を、リンク構造とページの文章内容からいかにうまく抽出できるかということである。そして、見つけ出したリンクの強さと役割、ページ間の結び付きから、コンテンツグループを発見する。将来的には HTML4.0[13] などによって役割を持ったリンクが普及して、リンクの意図の細かい抽出にそれらのリンクの属性を積極的に使うことが期待される。

Web ページ群のグループ化、階層化、順序付けを、より正確に行うためには、これらのリンクを形成する意図をなるべく Web ページから抽出することが必要になってくる。Web ページ制作者のリンク生成の意図を推定できれば、Web 利用者をよりうまく誘導できるようになるだろう。さらにリンクによる Web 情報の分類は、単語をキーとする分類などの他の分類方法と組み合わせることにより、より細かく Web 情報を分類するための助けとなる。

次節以降で、実際に細かくリンクの性質を取り扱う方法を述べる。

4.2 リンク構造からの分類

本節ではリンクの性質に基づいて、リンク構造情報を抽出して索引を作成し、リンク構造を使用した情報検索に利用する。この節で Web ページの間の関係をリンク構造から決定される「距離」と「影響度」という 2 つの尺度が定義される。

4.2.1 リンクの距離

前節で述べたようにページ間のリンクをたどる回数は、ページ間のつながりの強さと密接な関係がある。リンクはその Web ページ制作者にとって、リンクを作成するコストに見合う参照関係を示している。2 つの Web ページ間を考えた場合、その両ページのリン

クをたどる回数と、その両ページの関係の深さには相関関係が期待される。このリンクをたどる回数を 2 つのページの距離として定義する。

まず、2 つのページの間のリンクによる経路を定義する。ページ a からページ b への経路とは、ページ a から順にページのリンクをたどってページ b へ到達する、一連のページとリンクのまとまりである。

ページ $a \rightarrow$ ページ a 内のリンク $x \rightarrow$ ページ $i \rightarrow$ ページ i 内のリンク $y \rightarrow \cdots \rightarrow$ ページ b

経路は存在しない場合もあり、複数存在する場合もある。また、ループを構成している場合は無数に存在する。

2 つのページ間の距離は、その間の経路でリンクをたどる回数が最小のものと定義する。経路が存在しないときの距離は無限大である。

例えば図 4.1のページ A からページ E へは、ABDE という経路と、ADE という経路が存在するが、ADE の方が短いので距離 AC は 2 である。またページ F からページ A への経路は存在しないので、距離 FA は無限大である。

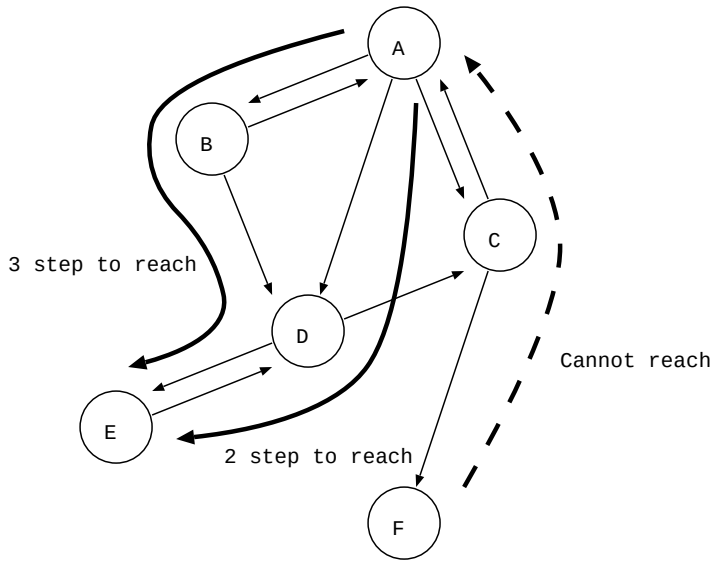

図 4.1: リンクの距離

以上から、$link(a \Rightarrow b)$ をページ a からページ b へのリンクを表すとすると、ページ a からページ b への距離 $distance(a \Rightarrow b)$ は、以下の式で定義される。

$a == b$ のとき

$$distance(a \Rightarrow b) = 0 \tag{4.1}$$

4.1以外で、a がリンクを持たないとき

$$distance(a \Rightarrow b) = \infty \tag{4.2}$$

4.1、4.2以外で、$link(a \Rightarrow b)$ が存在するとき

$$distance(a \Rightarrow b) = 1 \tag{4.3}$$

4.1、4.2、4.3以外のとき

$$distance(a \Rightarrow b) = min(distance(x \Rightarrow b) + 1)\ (x \in \{\forall x : link(a \Rightarrow x)\}) \tag{4.4}$$

リンクの距離は、2 つのページのグループ間の距離に拡張できる。
まず、$link(a \Rightarrow b)$ をグループ間の $Link(A \Rightarrow B)$ に拡張する。

$link(a \Rightarrow b)(\exists a, \exists b(a \in A, b \in B))$ が存在するとき

$$Link(A \Rightarrow B) = true \tag{4.5}$$

4.5以外のとき

$$Link(A \Rightarrow B) = false \tag{4.6}$$

すると、グループ A とグループ B との間の単方向距離 $Distance(A \Rightarrow B)$ は、以下の通りに定義される。

$a == b(\exists a, \exists b(a \in A, b \in B))$ が成り立つとき

$$Distance(A \Rightarrow B) = 0 \tag{4.7}$$

4.7以外で、A がリンクを持たないとき

$$distance(A \Rightarrow B) = \infty \tag{4.8}$$

4.7、4.8でなく、$Link(A \Rightarrow B)$ が存在するとき

$$Distance(A \Rightarrow B) = 1 \tag{4.9}$$

4.7、4.8、4.9以外のとき

$$Distance(A \Rightarrow B) = min(Distance(X \Rightarrow B) + 1)(X \in \{\forall X : Link(A \Rightarrow X)\}) \tag{4.10}$$

リンクを逆向きにたどる場合の距離も考えることができる。これを逆距離と呼ぶと、ページ A から B への逆距離 ($reverse\_distance$) は単純に B から A への距離と同じである。

つまり、

$$reverse\_distance(a \Rightarrow b) = distance(b \Rightarrow a) \tag{4.11}$$

である。

リンクの向きを無視して距離を考えることもできるが、基本的に Web において利用者は、リンクを逆向きにたどることはできないので、あまり意味のある距離とはいえず、本研究では採用しない。

4.2.2 リンクの強連結グループ

リンクの距離を定義したのに続いて、リンクの一定距離以内のページの強連結グループを定義する。

2 つの Web ページ間で、双方向にリンクをたどって行き来できる (両ページが強連結の関係にある) 場合には、両ページの制作者が、両ページにある程度の関連があることを合意していると見なすことができる。したがって、どの Web ページでも距離が一定の値以下であるリンク構造を持つ Web ページ群は、特に強力なつながりをもっていて、全体で 1 つの Web コンテンツグループとして取り扱うことができる。グループとする距離を変化させることで、近い距離の Web ページのみをグループにする絞り込みと、距離条件を緩める拡大検索を行うことも可能となる。

本研究では N-強連結の関係にある Web ページをまとめて Web コンテンツグループとみなす。

ページ A と B が強連結であるとは、ページ A から B へのリンクの経路、および、ページ B から A へのリンクの経路の両方が存在することである。

ページ A と B が N-強連結であるとは、

$$distance(a \Rightarrow b) + distance(b \Rightarrow a) \leq N \tag{4.12}$$

ということである。

N-強連結ページグループを構成するページは他のページにリンクをたどって戻ってくるのに必要なリンクをたどる回数が N 回以内なのである。別の言い方をすれば、N 以内の数のリンクによってページがループを構成しているグループと言える。近距離でループ

を構成していると言うことは、構成ページの作者は互いのページを認識している可能性が高いことを示している。

N をどの程度に取るかを変化させることは、近い距離の Web ページのみをグループにする絞り込みと、距離条件を緩める拡大検索を行うことを意味する。

例えば図 4.2のように、様々な強連結グループが存在している。また、距離を伸ばすにつれて、ページ AC からなるグループから ACD、ACDB、ACDBE へと拡大したり、逆に絞り込むことができる。

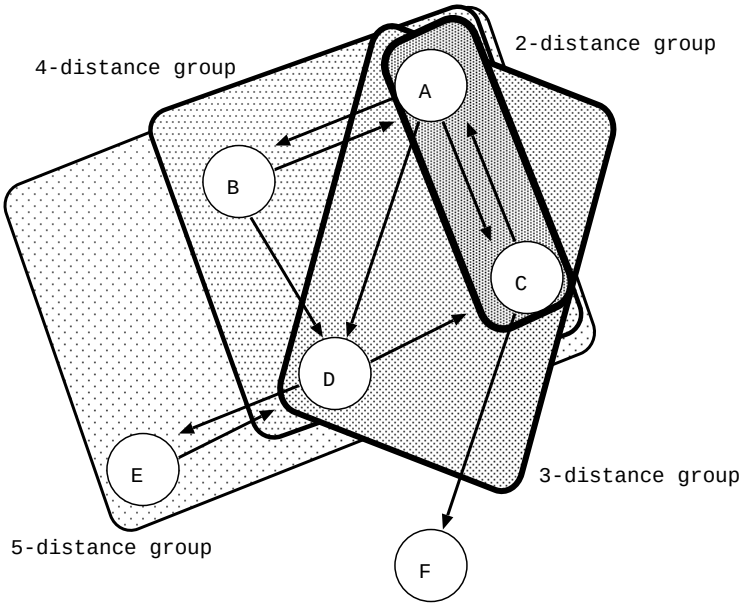

図 4.2: リンクの強連結グループ

4.2.3 リンクの影響度

図 4.3でのページ A と B のグループとページ D と E のグループは両方とも同じ距離 2 の強連結グループを構成している。しかし、利用者のページの閲覧をモデル化して、ページ内のリンクを無作為に選ぶ場合を考えると、ページ AB のグループから閲覧を開始した利用者が、ページ AB のグループ内にとどまる確率は、ページ DE のグループから開始してページ DE のグループにとどまる確率よりも、低い。なぜなら、ページ AB のグループの方がグループの外へでていくリンクの数の割合が高いからである。

これをリンクの数から見た場合は、ページ DE のグループのページはリンク数が少な

く、リンク 1 つ当りがページを結び付ける力は、ページ AB のグループのページより強いということになる。つまりリンクの数も、リンクの強さを推測する上での大きな要素である。リンクが少ないページのリンクは、リンクの多いページのリンクよりもたどられる確率が高い。つまり、リンク数が少ないページのリンク 1 つ当りがページを結び付ける力は、リンク数が多いページより強いということになる。

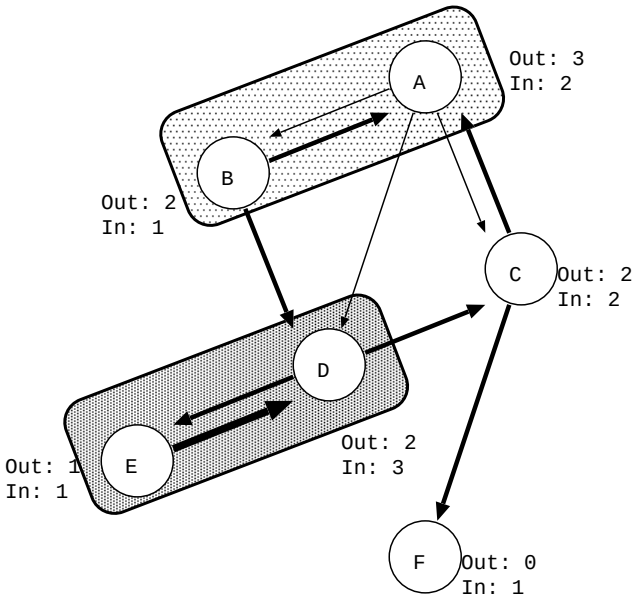

図 4.3: リンクの数

以上のことを反映する、リンクの影響度を定義する。

2 つのページ間のリンクによる影響度は、一方のページからリンクをたどった場合 (複数のリンクが存在する場合は等確率で 1 つのリンクを選ぶとして)、他方のページへたどり着く確率と定義する。ただし、同じページを複数回経由する経路はたどり着く確率から除外する。

各ページでリンクが複数ある場合、それらの中からリンクの種類に限らず、等確率で 1 つのリンクが選ばれると仮定する。また、利用者はリンクを使わない履歴のロールバックはしないものとも仮定している。目標のページにたどり着くか、最初のページに戻った場合もそれ以上の探索は行わない。

例えば図 4.4において、ページ C は出リンクを 2 つ持っている。等確率でリンクを選ぶと仮定するので、影響度 CA と影響度 CF は等しく $\frac{1}{2}$ である。また、ページ A からページ E へは、ABDE という経路を通る確率が $\frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{12}$ であり、ADE という経路を

通る確率が $\frac{1}{3} \times \frac{1}{2} = \frac{1}{6}$ であるので、合計の $\frac{1}{4}$ が影響度 AF である。

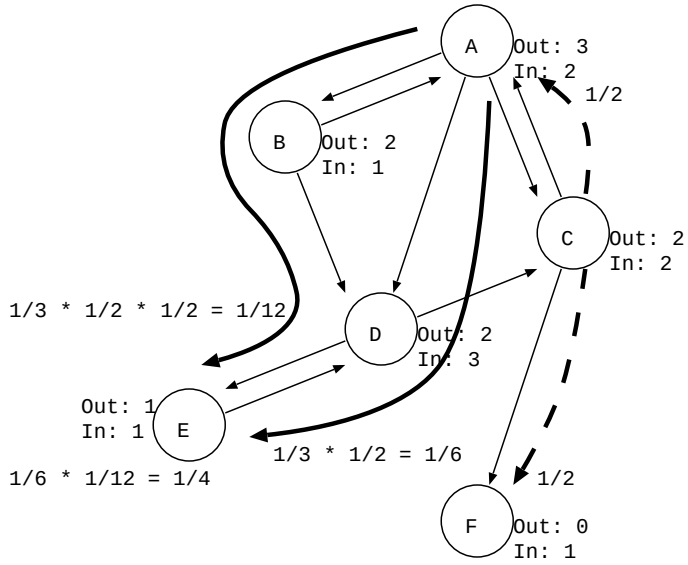

図 4.4: リンクの影響度

すると、$count_{link}(a)$ をページ a の総出リンク数として、ページ a からページ b への影響度 $influence(a \Rightarrow b)$ は以下の式で定義される。

$a == b$ のとき

$$influence(a \Rightarrow b) = 1 \tag{4.13}$$

4.13以外で、$distance(a \Rightarrow b) == \infty$ のとき

$$influence(a \Rightarrow b) = 0 \tag{4.14}$$

4.13、4.14以外のとき

$$influence(a \Rightarrow b) = \sum_{x \in \{\forall x : link(a \Rightarrow x)\}} \frac{influence(x \Rightarrow b)}{count_{link}(a)} \tag{4.15}$$

続いて、リンクの影響度を、2 つのページのグループ間の影響度に拡張する。

グループ A からグループ B への影響度 $Influence(A \Rightarrow B)$ は、$Count_{link}(A)$ をグループ A の外への総出リンク数として、以下の式で定義される。

$A \equiv B$ のとき

$$Influence(A \Rightarrow B) = 1 \tag{4.16}$$

4.16でなく、$Distance(A \Rightarrow B) == \infty$ のとき

$$Influence(A \Rightarrow B) = 0 \tag{4.17}$$

4.16、4.17以外のとき

$$Influence(A \Rightarrow B) = \sum_{X \in \{\forall X : Link(A \Rightarrow X)\}} \frac{Influence(X \Rightarrow B)}{Count_{link}(A)} \tag{4.18}$$

距離と同様に、リンクを逆向きにたどる場合の影響度も考える。これを逆影響度 ($reverse\_influence$) と呼ぶが、逆影響度は出発点となるページを見つけ出す際に特に重要である。ページ A から B への逆影響度は距離の代わりに逆距離、出リンクの代わりに入リンク用いて計算される。

つまり、

$a == b$ のとき

$$reverse\_influence(a \Rightarrow b) = 1 \tag{4.19}$$

4.19以外で、$reverse\_distance(a \Rightarrow b)(== distance(b \Rightarrow a)) == \infty$ のとき

$$reverse\_influence(a \Rightarrow b) = 0 \tag{4.20}$$

4.19、4.20以外のとき

$$reverse\_influence(a \Rightarrow b) = \sum_{x \in \{\forall x : link(x \Rightarrow a)\}} \frac{reverse\_influence(x \Rightarrow b)}{count_{reverse\_link}(a)} \tag{4.21}$$

である。

リンクの向きを無視して影響度を考えることは、距離の場合と同様に、Web ではあまり意味のあることとはいえず、本研究では扱わない。

4.2.4 リンクによる階層

影響度、被影響度を使うとさらに、リンク構造からみた階層性を数値としてとらえることができる。図 4.5に、各ページからの影響度の合計 (Sink) と被影響度の合計 (Source) を示す。被影響度の合計が優位なページ (A,B) はグループ内でより概略的な内容を示すページであり、逆に影響度の合計が優位なページ (E,F) はグループ内でより詳細な内容を示すページである。ページ A がもっとも被影響度の合計が高く、閲覧の開始ページとして最適なことがわかる。また、影響度の合計の絶対値で見るとページ D がもっとも高いので、ページ D がこのモデルでは利用者がもっとも頻繁に閲覧する可能性が高いことになる。

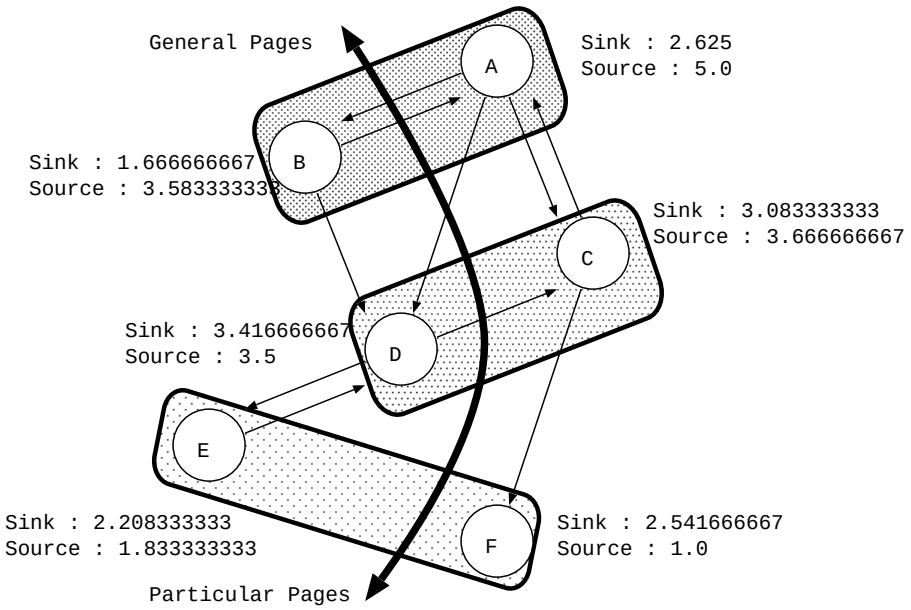

図 4.5: リンクによる階層

4.3 文章情報からのリンクの分類

この節では、言語学的な観点から、Web ページのリンクによる結び付きの強さや性格を見いだす方法を提案する。

4.3.1 語の出現率の利用

各ページの文章を調べて、その中に特徴的に登場する単語の出現頻度を考える。もし、リンクされているページ間に共通して頻繁に登場する単語が存在していれば、それはリンクされているページ間の結び付きを強める傍証と考えられる。

ある単語がどれくらい文章群全体の中で特徴的かを表すために、tf・idf 法によって、単語の重み付けをおこなう。tf・idf 法とは、文章群の中での、低頻度語の重要度を上げ、高頻度語の重要度を下げる方法である。あるページ P のある単語 w に対する重み付けの得点を score(P, w) とする。すると、

$$tf(P, w) = w \text{ が } P \text{ の中に出現する頻度} \tag{4.22}$$

$$idf(w) = log(\frac{\text{全文章数}}{w \text{ を含む文章数}}) \tag{4.23}$$

として、

$$score(P, w) = tf(P, w) \times idf(w) \tag{4.24}$$

と表すことができる。各単語の出現頻度 (tf、Term Frequency) に、全文章の中での珍しさの重み (idf、Inverse Document Frequency) をかけるわけである。さらに、文章間の単語の量の違いによる差をなくすために、文章の全単語の量によって正規化をはかる。両ページに共通して出現する各単語に対して、両ページの得点の相乗平均を求め、共通して出現する単語全てでの和を取ったものが、2 つのページの文章間の類似度である。つまり 2 つのページ P、Q の類似度を similarity(P, Q) とおくと、類似度は次の式で表される。

$$similarity(P, Q) = \sum_{w \in P \text{ と } Q \text{ に共通に出現する単語}} \sqrt{\frac{score(P, w)}{P \text{ の単語数}} \times \frac{score(Q, w)}{Q \text{ の単語数}}} \tag{4.25}$$

相加平均ではなく、相乗平均を使っているのは、両方のページに共通してある一定以上の割合で登場する単語を重視するためである。

4.3.2 文章構造の利用

Web ページの文章から得られる情報から、より詳しくリンクの性質を分析して、Web ページ間の関係の数値化を試みる。

HTML を細かく分解すると、識別子としての URL、文章、制御用のタグ (特にリンク) などの要素により構成されている。これらの要素をリンクの情報になるべく反映させることが目的である。HTML の構造からみたリンクの位置や、リンクの周囲のキーとなる単語を調べるといった方法を採り入れることにより、より細かく Web 情報を分類するための助けとなる。

本研究では、Web ページ間の関係を表すために「関連度」、「抽象度」、「相対順序」の 3 つの数値を導入する。

関連度

関連度は複数の文章の間の関連の度合の尺度である。ある 2 つの Web ページ間の関連度がの高いことは、周りの Web ページと比べて相対的に、その 2 つの Web ページ間の結び付きが強いことを示す値である。

Web ページ a と Web ページ b の間の (単方向) 関連度を $relation(a \Rightarrow b)$ で表すとする。表 4.1はある Web ページ (これを a としている。) 内の様々な要素から、いくつかの Web ページ間の関連度が導き出されることを示している。表の「関連度の変化」の中の大小は、他の要素と比較して相対的に、その要素がどの程度の強い関連を示しているかということである。例えば、「リストで列記されている」要素による 2 つの Web ページの関連は、「ディレクトリ関係を参照する」要素による関連よりも相対的に強いので、「リストで列記されている」要素による変化は中、「ディレクトリ関係を参照する」要素による変化は小ということである。

対象となる Web ページ群を解析して、各ページ間の関連度を算出する。2 つの Web ページ間に、要素が複数あったり重複した場合は、それらの要素による関連度を累積したものが最終的な関連度となる。結果として関連度は、(Web ページ間が全く無関連であることを示す)0 以上の値を取る。

双方向関連度は 2 つの単方向関連度の相乗平均に類似度を加えたものでで、Web ページ a と Web ページ b の間の双方向関連度を $relation(a \Leftrightarrow b)$ とすると、以下の式で表される。

$$relation(a \Leftrightarrow b) = similarity(a, b) + \sqrt{relation(a \Rightarrow b) \times relation(b \Rightarrow a)} \qquad (4.26)$$

関連度を使って結び付きの強い Web ページ群のグループを見つけ出すことができる。5.2節では、この関連度を利用した Web ページ群のグループ化を行う。

抽象度

ある 2 つの Web ページ間で比較した場合、一方の Web ページが他方も Web ページより相対的に、抽象的あるいは具体的な内容を表しているかどうかを、抽象度は示す。

2 つの Web ページ間の (単方向) 抽象度 ($reification(a \Rightarrow b)$ で表す) は、Web ページのリンクが指し示す先が、そのページの内容より抽象的 (あるいは具体的) な Web ページかどうかを、表 4.2に挙げられている Web ページの要素から決定する。表の「抽象度の変化」部分の大小は、関連度の場合と同様に、他の要素と比較して相対的に、抽象度にどの程度の影響を与えるかということである。2 つの Web ページ間に、要素が複数あったり重複した場合は、それらの要素による抽象度を累積したものが最終的な抽象度となる。

抽象度は正負両方の値を取る。$reification(a \Rightarrow b)$ が大きい正の値を持つときは、b は a の内容をより具体的にした内容を持っていることを示す。逆に大きい負の値を持つとき

[1]直接のリンクには影響度に加えてボーナスを与えている

要素	例	関連度の変化
影響度 ($influence(a \Rightarrow b)$)	4.2節参照	$relation(a \Rightarrow b)+ =$ 大 (影響度に比例)
リンクが直接張られている[1]	<A HREF=“b”> $\cdots$ </A>	$relation(a \Rightarrow b)+ =$ 大
リンクの性質	<LINK rel=“alternative” href=“b”>[13]	$relation(a \Rightarrow b)+ =$ 大 (種類で変化)
タグによる強弱	<H2><A HREF=“b”> $\cdots$ </A></H2>	$relation(a \Rightarrow b)\pm =$ タグの強さによる修正
リストで列記されている	<UL> <LI><A HREF=“b”> $\cdots$ </A> <LI><A HREF=“c”> $\cdots$ </A> </UL>	$relation(b \Rightarrow c)+ =$ 中 $relation(c \Rightarrow b)+ =$ 中
同じ段落で列記されている	<P> <A HREF=“b”> $\cdots$ </A> <A HREF=“c”> $\cdots$ </A> </P>	$relation(b \Rightarrow c)+ =$ 中 $relation(c \Rightarrow b)+ =$ 中
ディレクトリ関係を参照する	a = http://SAMEDIR/a.html b = http://SAMEDIR/b.html	$relation(a \Rightarrow b)+ =$ 小 $relation(b \Rightarrow a)+ =$ 小
文章の内容の類似性 [16]	a = http://ORIGINAL/a.html b = http://MIRROR/a.html	$relation(a \Rightarrow b)+ =$ 小 $relation(b \Rightarrow a)+ =$ 小

表 4.1: 関連度の要素 (Web ページ “a” を調べる場合)

はbはaの内容をより抽象的、メタにした内容を持っている。0に近い値の場合は、a、b両者は対等な関係の内容を持っているか、無関係 (このときは関連度も低い) かである。

双方向抽象度は2つの単方向抽象度の差 (双方向関連度と異なり、和ではない) で、Webページaと Webページbの間の双方向抽象度を $reification(a \rightleftharpoons b)$ とすると、以下の式で表される。

$$reification(a \rightleftharpoons b) = reification(a \Rightarrow b) - reification(b \Rightarrow a) \tag{4.27}$$

抽象度は、関連度によってグループ化された Webページ群の階層化に用いる。階層化に付いては、6.3節で述べる。

相対順序

相対順序は2つの Webページの前後関係を示す値である。主に関連度や抽象度を補佐するものとして用いられる。

2つの Webページ間の (単方向) 相対順序 ($order(a \Rightarrow b)$ で表す) は、Webページのリンクが指し示す先が、そのページの内容より前に来る部分のページかどうかを、表4.3にある Webページの要素から決定する。表の「相対順序の変化」部分の大小は、関連度の場合と同様に、他の要素と比較して相対的に、相対順序にどの程度の影響を与えるかということである。2つの Webページ間に、要素が複数あったり重複した場合は、それらの要素による相対順序を累積したものが最終的な相対順序となる。

相対順序は正負両方の値を取る。$order(a \Rightarrow b)$ が大きい正の値を持つときは、aの内容はbの内容より前の順番に来ることを示す。逆に大きい負の値を持つときはbがaより前に来る。0に近い値の場合は、a、b両者は順不同であるか、無関係 (このときは関連度も低い) かである。

双方向相対順序は2つの双方向抽象度と同様に単方向相対順序の差で、Webページ Aと Webページ Bの間の双方向相対順序を $order(a \rightleftharpoons b)$ とすると、以下の式で表される。

$$order(a \rightleftharpoons b) = order(a \Rightarrow b) - order(b \Rightarrow a) \tag{4.28}$$

相対順序は、関連度、抽象度を補佐して、グループ化された Webページ群の階層化、順序付けに用いる。階層化、順序付けに付いては、6.3 節で述べる。

要素	例	抽象度の変化
リンクの性質	<LINK rel=“Contents” href=“b”> <LINK rel=“Index” href=“b”>	$reification(a \Rightarrow b)- =$ 大 $reification(b \Rightarrow a)+ =$ 大 (種類で変化)
リンク (及びその前後) の単語の情報	<A HREF=“b”> 目次 </A> <A HREF=“b”> 索引 </A>	$reification(a \Rightarrow b)- =$ 大 $reification(b \Rightarrow a)+ =$ 大 (単語で変化)
違う深さのリストで列記されている	<UL> <LI><A HREF=“b”> $\cdots$ </A> <UL> <LI><A HREF=“c”> $\cdots$ </A> </UL> </UL>	$reification(b \Rightarrow c)+ =$ 中 $reification(c \Rightarrow b)- =$ 中
違う深さの段落で列記されている	<P> <A HREF=“b”> $\cdots$ </A> <P> <A HREF=“c”> $\cdots$ </A> </P> </P>	$reification(b \Rightarrow c)+ =$ 中 $reification(c \Rightarrow b)- =$ 中
ディレクトリ関係を参照する	a = http://SAME/a.html b = http://SAME/SUB/b.html	$reification(a \Rightarrow b)+ =$ 小 $reification(b \Rightarrow a)- =$ 小
ファイル名に index などが使われている	a = http://SAME/index.html b = http://SAME/other.html	$reification(a \Rightarrow b)+ =$ 小 $reification(b \Rightarrow a)- =$ 小

表 4.2: 抽象度の要素 (Web ページ “a” を調べる場合)

要素	例	相対順序の変化
リンクの性質	<LINK rel=“Prev” href=“b”> <LINK rel=“Start” href=“b”>	$order(a \Rightarrow b)- =$ 大 $order(b \Rightarrow a)+ =$ 大 (種類で変化)
リンク (及びその前後) の単語の情報	<A HREF=“b”> 前 </A> <A HREF=“b”> 最初 </A>	$order(a \Rightarrow b)- =$ 大 $order(b \Rightarrow a)+ =$ 大 (単語で変化)
順序付きリストで列記されている	<OL> <LI><A HREF=“b”> $\cdots$ </A> <LI><A HREF=“c”> $\cdots$ </A> </OL>	$order(b \Rightarrow c)+ =$ 中 $order(c \Rightarrow b)- =$ 中
ファイル名に数字などが使われている	a = http://SAMEURL/1.html b = http://SAMEURL/2.html	$order(a \Rightarrow b)+ =$ 小 $order(b \Rightarrow a)- =$ 小

表 4.3: 相対順序の要素 (Web ページ “a” を調べる場合)

第 5 章

リンク構造情報を用いたグループ化による情報検索

前章のリンクの性質に基づいて、本章ではリンク構造を使用したコンテンツグループに基づく情報検索をおこなう。リンクを通して情報に重みを付け、グループ化を行い、情報検索に利用する具体的な方法を提案する。

5.1 リンクの距離に基づく強連結性を利用した、検索結果の圧縮

ここでは既存の全文検索エンジンの検索結果を距離 N 以内の強連結ページグループのコンテンツグループへと圧縮する方法を提案する。

既存の全文検索エンジンによるキーワードの検索によって、ヒットしたページとその得点をコンテンツグループごとに計算し直す。キーワードの出現頻度に基づくページの得点と近距離の強連結ページによって、コンテンツグループを求めるのである。

複数の Web ページで構成された情報は、ばらばらに提示されるために冗長であり、1 つのページにまとまっている場合よりも各ページの情報量は少ないので、検索結果の上位に入りにくい。そこで単に Web ページ単位で検索を行うだけでなく、リンクによって結果内の冗長な Web ページのグループを Web コンテンツグループとして検出して、結果の縮退をして冗長度を下げる。また、複数の Web ページにまたがる情報は、グループ単位で評価値を算出する。これらの方法により、利用者が手数以内の部分 (検索結果の上位部分) の適合率が高まり、より効率的な検索が可能となる。結果を修正する方式であるの

で、既存の全文検索エンジンとの親和性が高いのも特長の 1 つである。

5.1.1 グループの発見

まず、キーワードによる検索の結果から各ページの得点を全文検索エンジンによって求める。続いて、ページ毎にの距離 N 以内の強連結ページグループを見つけていく。強連結ページを見つけるには、以下のような幅優先の Dijkstra のアルゴリズムに基づいて、ページの探索をおこなえば良い。

```
queue = [‘p’]
result = []
dist = 1
while (dist < n) {
        newqueue = []
        for all ‘x’ (‘x’ is an outgoing link of queued pages) {
                push x in newqueue
                if ‘x’ == ‘p’ {
                        push x in result
                }
        }
        queue = newqueue
        dist += 1
}
return result
```

図 5.1: ページ P の距離 N 以内の強連結ページを見つけるアルゴリズムの概略

この結果得られた強連結ページグループに含まれるページの得点の和を求める。全文検索エンジンによるページ p の得点を score(p) とすると、ある強連結ページグループ G の得点 Score(G) は

$$Score(G) = \sum_{g \in G} score(g) \tag{5.1}$$

となる。

5.1.2 グループの大きさ

さらに、適切なグループの大きさを自動的に求める。つまり適切な強連結の距離を自動的に求める。

距離 2 の強連結グループから順々に距離を拡大して、各強連結グループの構成ページの得点の和を再計算する。このとき、距離が遠くなるほど利用者がグループの全容を把握するのが困難になるので、次の式のように距離に応じて前述の得点の和を減少させている。

$$Score(G) = W^n \times \sum_{g \in G}(score(g)) \tag{5.2}$$

W は酔歩確率 (Walk Rate) の意味で 0 から 1 までの値を取る。W は利用者がリンクをたどる確率を示している。つまり平均的な利用者の行動確率を

- 直接 URI を指定してページを取得する確率：1 - W
- リンクをたどる確率：W

と仮定したモデルに基づいている。実際には W はリンクの存在するページの内容に大きく左右されるが、各ページが適切なものであれば、1 に近いと考えられる。現在は W=0.8 として計算している。将来的には、ページの内容を判断して、適切な値を動的に求められるようにしたい。各グループの得点はその Web グループの正当性の高さを示し、高い得点のグループが適切な大きさの Web グループとして扱われる。

グループに含まれるページの数自身もリンクをたどる回数の大まかな目安となる。上位 H 位以内の適合率を保ちつつ情報量を最大にするという観点からは全ページ数 (N とおく) の H 分の 1 のページ数へと線形減少する関数で距離 n のグループ G_n に新たに加わるページの得点を以下のように減少させる。

$$Score(G_n) = W \times (Score(G_{n-1}) + (1 - \frac{H \times |G_n|}{N}) \times \sum_{g \in G_n - G_{n-1}} score(g)) \tag{5.3}$$

5.1.3 Web グループの遍在性

表 5.1は西暦 2000 年 1 月 25 日現在の www.jaist.ac.jp で提供される Web ページ (学内のみ閲覧可能なページや個人のページを除く。総ページ数 6,856。) にどれだけの Web グループが存在するかを調べたものである。例えば距離 5 の Web グループは 759 存在している。それらの Web グループは 2,317 のページから構成されている。これは全ページの

33.8%が Web グループを構成しており、Web グループの平均ページ数は 3 である。また、残りの Web ページを Web グループと考えると平均ページ数は 1.3 となる。

Web グループを強連結グループではなく、単にリンクされているものをグループにした場合には、2,014 の Web グループになるので、平均のページ数は 3.4 となる。

距離	Web グループ数	Web グループを構成しているページ数	残りのページ数
2	835	2,222	4,634
3	872	2,297	4,559
4	872	2,314	4,542
5	759	2,317	4,539

表 5.1: JAIST の Web グループ

一切リンクを持たないページが 2,508。一切リンクを受けないページが 2,646(古い情報などの理由でリンクを外されている、あるいは、Web サーバのディレクトリ閲覧機能を利用している)。また、JAIST 内の他のページに一切のリンクを持たず、また、リンクもされていないページ数は 1,937 である。

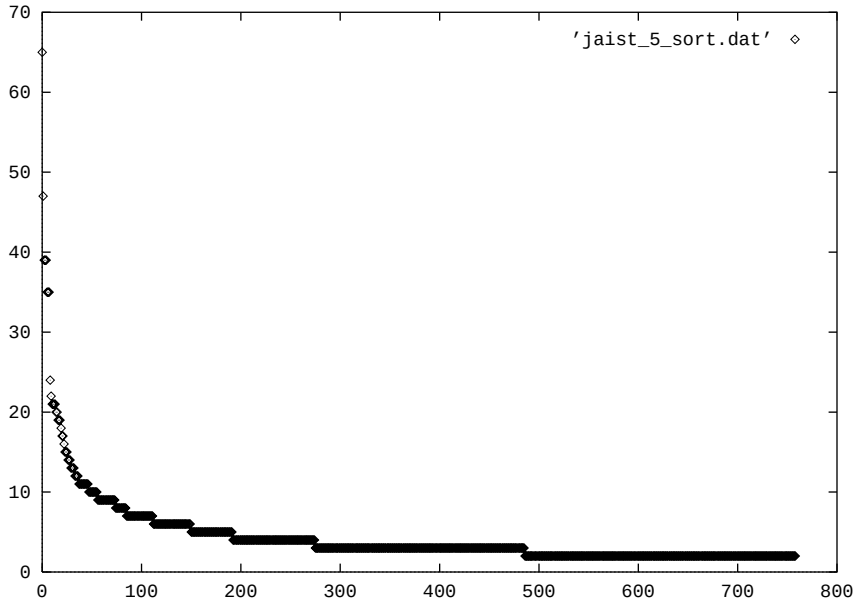

図 5.2: JAIST のグループの大きさ

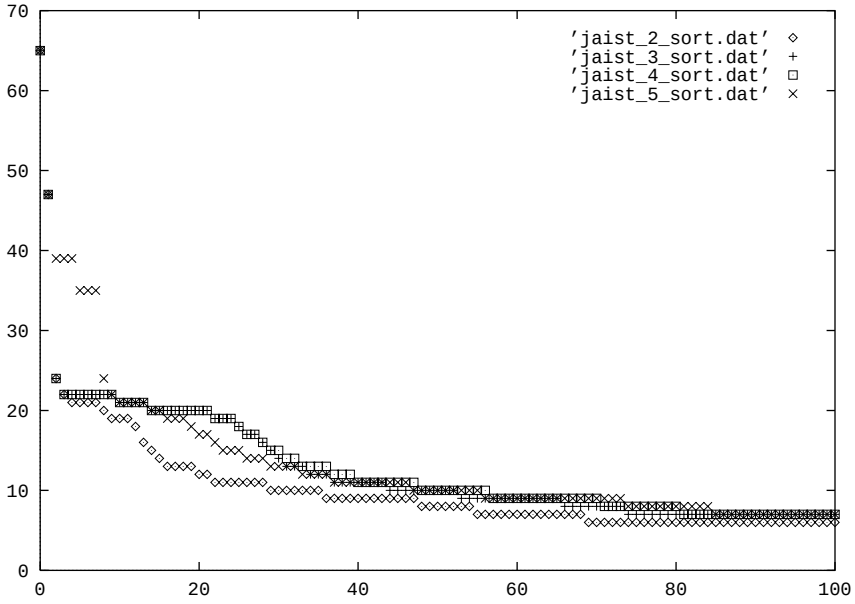

図 5.3: 距離による JAIST のグループの大きさの違い

実際に JAIST の情報科学科の研究室の紹介の部分を調べてみると、図 5.4のように篠田研究室のページと情報科学科のページがそれぞれ距離 2 の強連結グループを構成している。さらにこれらのグループは同じ距離 3 の強連結グループに含まれている。近距離の強連結グループを構成しているページが互いに補完しているページであることがわかる。したがって、本研究では近距離の強連結グループを Web グループとして取り扱い、それらのページをつなげているリンクの強さを他のリンクよりも強いものとして扱う。また、距離が強連結の距離が近ければ近いほどその結び付きは強いものとなる。

キーワード検索の結果の間で、2-5 程度の双方向距離以内の強連結グループは多く見られる。それらは、多くの企業サイト、オンラインマニュアルや FAQ、ツールで作成された一連の資料など、特にある程度のオーソリティを持ったものに良く見られる。

このような結果が適用できる例は、LaTeX2HTML や、PowerPoint などのツールで作られている一連の資料や、Web の上の FAQ、マニュアル、メーリングリストなど、多岐にわたる。

これらのオーソリティを持ったコンテンツグループは、目次などのコンテンツグループの影響を集約するページを持っている。そのため、影響度を使ってそれらの目次を、代表コンテンツとして選びだすことができる。

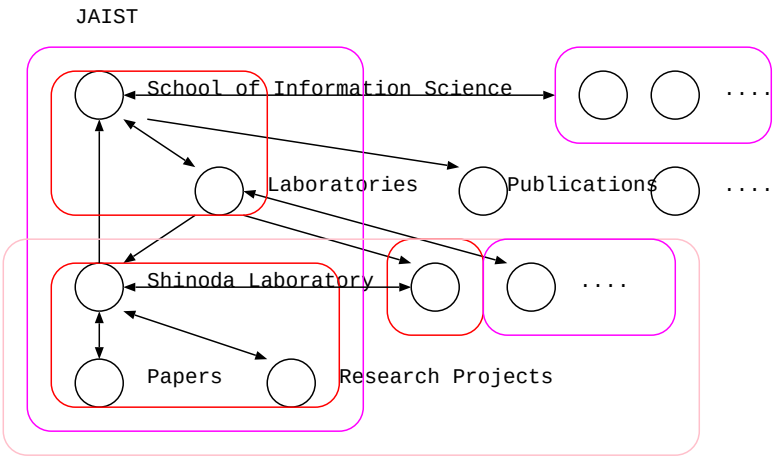

図 5.4: JAIST のグループ

5.2 関連度による検索結果の絞り込み、拡大

強連結によるグループ化よりもさらに、Web ページの内容を考慮したグループ化を行うために、4.3.2節の Web ページ間のリンクされた結び付きの強さを表す関連度を使用する。

Web ページ群の中から高い関連度でつながっている Web ページ群を見つけ出して、グループとする。求めるグループを G とおくと、$THRESHOLD$ を 0 以上の閾値を表す定数として、G は以下の式をみたす Web ページ群である。

$$G : relation(a \Leftrightarrow b) \geq THRESHOLD \ \ (\forall a.\forall b \in G) \tag{5.4}$$

実際に、ある Web ページ a から、a を含む閾値 THRESHOLD 以上のグループである G は、図 5.5のアルゴリズムによって見つけ出すことができる。

関連度では、リンクの距離だけでなく、リンクの影響度や Web ページの構成などから得られる情報が累積されており、$THRESHOLD$ を変化させる絞り込み検索と拡大検索は、距離による強連結グループと比べて、より細かく調整できる。

利用者に検索結果を提示するときには、連続的に閾値を変化させた結果を、関連度の密度を等高線図のようにして表すことができる。逆に利用者が一定の大きさのグループを望むのなら、そうなるように逆に閾値を動的に決定することができるという利点も存在する。

利用者側のインタフェースとして、閾値を連続して変化させる機能を導入すれば、連続した絞り込みや拡大検索や、利用者の望む大きさのグループを取り出す検索も可能と

```
  G = S = [a]
  while (b = pop(S)) {
    if (relation(x ⇔ y) >= THRESHOLD) (x ∈ {∀ x : link(b ⇒ x)}, ∀
y ∈ G)
      and !(x ∈ G) {
        push(G, x)
        push(S, x)
    }
  }
```

THRESHOLD	グループ化する関連度の閾値
G	ページ a から関連度が THRESHOLD 以上のであるグループ
S	未評価のページを保持するためのスタック

図 5.5: Web ページ a を含む、強い関連度を持つグループ探索のアルゴリズム

なる。

5.3 リンク構造を反映する「色」を用いた潜在的な共通グループの発見

ある Web ページがいくつかの異なる Web ページからそれぞれ、どのくらいリンクの影響を受けているかを考える。

例えば、お互いにリンクでは到達できない 2 つの Web ページであっても、多くの Web ページから共通にリンクされていれば、関連性の高い Web ページである。他のページからの影響を、距離と影響度に基づいて求めて、似たような傾向を持つ Web ページのグループを見つけ出すことができる。

ここでは従来のキーワードによる検索と組み合わせて、得られたリンク情報を利用者に分かりやすいように着色してグループ化した提示をすることにより、検索の効率の向上を図る。リンクの距離や数に応じて、Web ページに色をつけることにより、ある 2 つの Web ページ間のリンクによる関係を直感的に把握しやすくなり、検索結果の冗長度を取り除く方法に利用できる。

具体的には、検索結果周辺の Web 空間を HSV 表色系の色空間にリンク構造をマッピングする。HSV 表色系は、心理的に把握が容易なマンセル表色系に近く、また、RGB 表色系との変換が容易という利点がある。

キーワード検索の結果から、着色の根となる 1 つ、または、複数の Web ページを選択してから、残りの Web ページを各根からの距離にしたがって着色を行う。

キーワードによる検索結果の上位の Web ページから順に、代表 Web ページとして、まず、各代表 Web ページに元となる色を割り当てる。現在の実装における代表ページの色付けでは、代表ページの数で Hue(色相) を自動的に分割して、色空間全体にまんべんなくマップできるようにしている。Saturation (彩度) は最高値、Value(明度) には中央値を割り当てる。

続いて、各代表 Web ページを根とする最短到達木をグラフ上で考え、その木にしたがって、各代表 Web ページの色より、残りの Web ページの色付けを行う。出力リンクの数にしたがって、Saturation を分割して落し、また、リンクをたどった距離に応じて Value が低下する。

複数の異なる色が割り当てられるページでは、それらの色の混色を行う、混色のアルゴリズムは以下の通りである。

A から距離 N にある B の色は

$$Hue(B) = Hue(A) \tag{5.5}$$

$$Saturation(B) = Saturation(A) \tag{5.6}$$

$$Value(B) = Value(A) - W_N \times N \tag{5.7}$$

ただし W_N は距離 N における重み

色 C_A と色 C_B とを混色した色 C_N は

$$Red(C_N) = max(Red(C_A), Red(C_B)) \tag{5.8}$$

$$Green(C_N) = max(Green(C_A), Green(C_B)) \tag{5.9}$$

$$Blue(C_N) = max(Blue(C_A), Blue(C_B)) \tag{5.10}$$

で行なっている。

以上の方法を適用して、得られる着色から色が近い Web ページごとにグループにすることにより、リンク構造によって Web ページのグループを取り出すことができる。これ

らの結果、直接のリンク接続が乏しいグループでも、ユーザが状態の把握をしやすい結果を得られるようになる。

また、検索エンジンでキーワードを 2、3 個指定した場合、各キーワードに対応する領域に異なる色を割り振っていくと、各キーワードの頻度の割合が反映された色が各 Web ページにつけられていき、キーワードの重みの違いが検索結果に反映されるようにすることができる。

例えば、「金沢」の検索結果に「青」、「観光」の検索結果に「赤」を割り当てて着色をする。「金沢の観光」の Web ページは「紫」のグループ、「金沢」が主で「観光」が従のページは「青紫」のグループ、「観光」が主で「金沢」が従のページは「赤紫」のグループとして取り出す。このようにして Web 空間を各キーワードの影響している領域、そしてそれらの領域の重なり具合いや割合を利用者は見てとることが可能になる。

根となる色や、グループ化する閾値は現在は経験的に決定しているが、利用者側のインタフェースとして、これらの値を連続して変化させるスライダーを導入することにより、根となる色の変化でキーワードの重み付けが実現し、グループ化する閾値を変化させることで、連続した絞り込みや拡大検索が可能となる

第 6 章

リンク構造情報を用いた階層化、順序付け

リンクを通した情報に重みを付け、階層化、順序付けする方法を検証する。この節での方法は単独で Web ページ群に適用するだけでなく、前節でのいずれかの方法でグループ化を行ったあとに、グループ間、あるいは、グループ内で階層化、順序付けにも使用する。

6.1 リンクの距離、影響度による検索結果の修正

5.1節において、検索結果の圧縮を行った。この見つかったグループを、実際に圧縮して利用者に提供する場合、このグループのどの Web ページの URL を利用者に提供するか選択する必要がある。

検索の結果から実際の情報を調べる際には、いくつかのリンクをたどって全体を把握することが多い。複数の Web ページで構成された Web コンテンツグループが、木構造 (あるいはそれに近い構造) をしているときに、葉のページのみを検索の結果として提示されると、全体の把握が難しくなる。

この結果内の吟味を効率的にするためには、リンクの影響度を利用して、利用者がグループを調べる際にリンクをたどる回数をなるべく減らす。Web コンテンツグループからグループを把握しやすい代表となるページ (例えば木構造の根) となる Web ページを Web グループの閲覧の開始ページとして、グループから検出して結果に提示する。

従来の検索エンジンでは、一般的に検索結果の上位から結果へのリンクをたどり、その先でさらにいくつかのリンクをたどる。リンクされたページは、検索の結果として (リンクを考慮しない場合に比べて) より望ましいページである可能性が高い。各ページのキーワード検索の得点を、そのページからの影響度に応じて、加えることにより、結果得られる得点で上位のページを、その Web コンテンツグループを代表するページとして選定

する。

代表ページ以外の Web グループのページを検索結果から除去することで、検索結果の情報量を増やす。

6.1.1 逆影響度の計算

キーワードによる検索の結果から各ページの得点を全文検索エンジンによって求める。続いて、ページ毎にの距離 N 以内の逆影響度を求める。ある逆影響度を求めるには、以下のような幅優先の Dijkstra のアルゴリズムに基づいて、逆影響度の伝搬をおこなえば良い。

```
for all page x {
        result[‘x’] = 0
}
queue = [([‘p’], 1.0)]
dist = 1
while (dist < n) {
        newqueue = []
        for all (path, infl) in queue {
                #l is incoming link number of a path’s last page
                for all ‘x’ (‘x’ is an incoming link of a path’s last page) {
                        continue if (path includes x)
                        result[‘x’] += (infl / #l)
                        push ([path, x], infl / #l) in newqueue
                }
        }
        queue = newqueue
        dist += 1
}
return result
```

図 6.1: ページ’p’ の他ページへの被影響度の伝搬を調べるアルゴリズムの概観

各ページの得点を上記のアルゴリズムから求めた被影響度と基づいて他のページに伝

搬させ、それらの和を求める。結果として、ページ p の新しい得点 newscore(p) は以下の式によって求められる。

$$newscore(p) = \sum_{g \in G}(score(g) \times reverse\_influence(g, p)) \tag{6.1}$$

この計算によってもっとも高得点のページが、適切なが代表ページとして選択される。

6.1.2 距離の影響

さらに、距離による利用者の閲覧の中止の可能性を考えると、上記の式は、距離の修正を加えて以下のようになる。

$$newscore(p) = \sum_{g \in G}(score(g) \times reverse\_influence(g, p) \times W^{distance(p,g)}) \tag{6.2}$$

W は 5.1節の酔歩確率である。

6.2 リンクによる階層

実際に JAIST の情報科学科の研究室の紹介の部分を調べてみると、図 5.4のように出リンクが入リンクより優勢な篠田研究室のページは周囲のページより概略的な内容のページであり、篠田研究室のグループのなかでは高い得点を示すページである。

6.3 抽象度、相対順序による検索結果の階層化、順序付け

いくつかのグループがさらにまとまって大きなグループを形成している場合もある。4.2節の抽象度、および、4.3節の相対順序を使って、それらの Web ページ (群) のグループ間の階層化、順序付けを行う。

関連度を使ってグループ化を施した後に、グループ間、及び、グループ内での階層化、順序付けを行なっているが、逆に先に階層化を行ない。その結果でグループ化も可能である。

基本的に抽象度でおおざっぱな階層構造を見つけ出し、抽象度では同じとみなされる細かい範囲で相対順序を適用していく。

双方向の抽象度、あるいは、相対順序で、あまり差の無い範囲の Web ページ (群) は同一と見なしつつ、一定以上の差が見られる Web ページ (群) の間でトポロジカルソートを行なっている。

どの程度の差までを同一と見なすか、その閾値を変えながら、繰り返しソートを適用することで、順々に拡大、あるいは、絞り込んだ階層化をすることができる。

以下では、5.2節と同じ例で、表 6.1のパラメータを与えて、抽象度を計算した結果が表 6.2である。

図 6.2の濃い矢印が、双方向抽象度を図示したもので、i は 他の Web ページに対してより抽象的、メタな内容を持っており、i と他の Web ページで階層を分けることができる。

要素	抽象度の変化
リンク (及びその前後) の単語の情報	5
下位のリストに記されている	± 2
下位の段落で列記されている	± 2
ディレクトリ関係を参照する	± 1
ファイル名に index などが使われている	± 1

表 6.1: 抽象度判定のパラメータ

	i	c	s	f
i	0	9	3	5
c	-11	0	0	0
s	-11	0	0	0
f	-11	0	0	0

表 6.2: 抽象度の計算結果

続いて、相対順序を表 6.3のパラメータを与えて、計算した結果が表 6.4である。

図 6.2の薄い矢印が、双方向抽象度を図示したもので、s、f、c の順序付けが下位のグループ内で行われる。

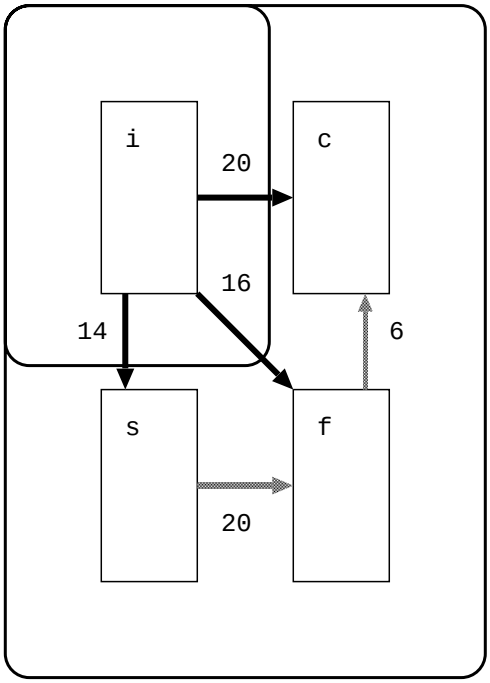

図 6.2: 双方向抽象度、相対順序による階層化、順序付け

要素	相対順序の変化
リンク (及びその前後) の単語の情報	± 5
順序付きリストで列記されている	± 2
ファイル名に数字などが使われている	± 1

表 6.3: 相対順序判定のパラメータ

	i	c	s	f
i	0	-2	-1	-2
c	2	0	0	-3
s	1	0	0	10
f	2	3	-10	0

表 6.4: 相対順序の計算結果

第 7 章

Web 上のリンク構造を利用した情報検索の実現方法

本論文でいままでに提案した方法の実験・評価のために、プロトタイプの設計と実装を行い。それから得られた結果を元に考察を行う

7.1 プロトタイプ

プロトタイプ 'Namazu-Hige'(仮称) の概要を図 7.1に示す。図の左側が全文検索部である Namazu 部であり、右側がリンク検索部である Hige 部である。

現在のプロトタイプは、5.1節の「リンクの距離に基づく強連結性を利用した、検索結果の圧縮」と#節の「リンクの距離、影響度による検索結果の修正」の部分を実装したものである。

図には表れていないが、HTML ファイルを収集するロボットとしては wget を利用している。ロボット (wget) によって得られた html ファイル群を大本のデータとしている。

7.1.1 全文検索部

全文検索部には、既存の全文検索システム Namazu を採用している。Namazu を採用した理由は以下の通り。

- オープンソースである。
- C と Perl で書かれていて、多くのプラットフォームで手軽に動作する。

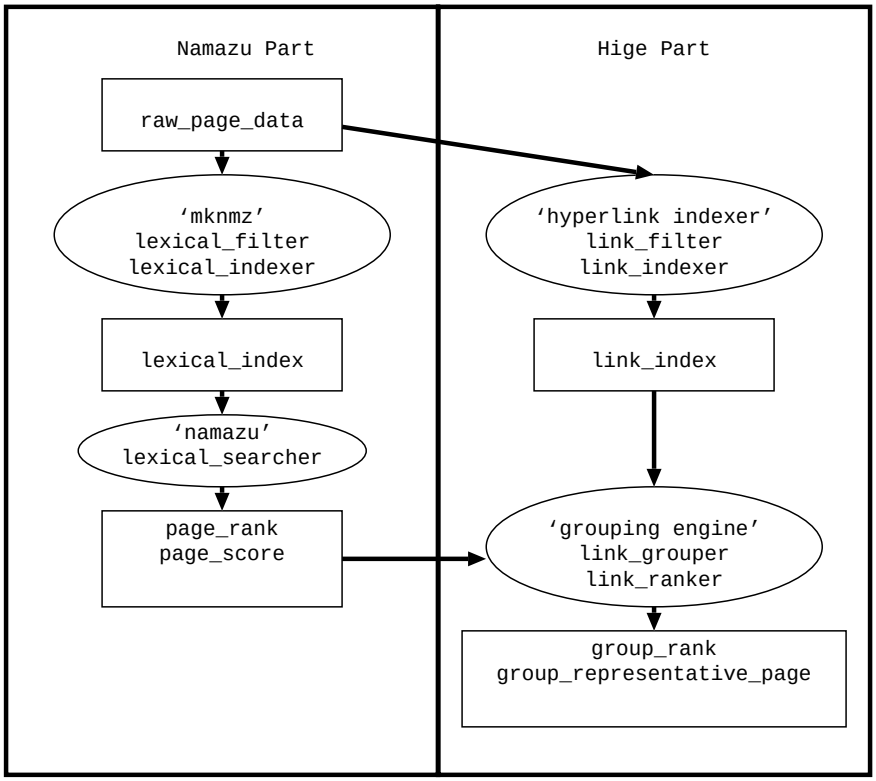

図 7.1: プロトタイプの概要

- 日本語コードを取り扱うことができる。
- 日本を主として広く使われている。

全文検索部は大きく分けて、文章のフィルタ・インデックス化をおこなう mknmz 部と、検索をおこなう namazu 部からなる。mknmz 部は文章を解析して語の出現数を数えると同時に、文章構造に基づいて語の重み付けをおこなう。それらの結果は高速に検索できるようにインデックスにまとめられて保存される。namazu 部は利用者が与えたキーワードを元にインデックスを検索して各文章の得点を見つけ出す。キーワードが複数与えられた場合は、tf・idf 法によってキーワード間の重み付けをおこなっている。結果として高得点順に文章を利用者に提示する。

7.1.2 リンク検索部

リンク検索部が本プロトタイプの主要部分である。本部分は Ruby で実装されている。全文検索部と同様に、リンク情報のインデックス化をおこなう部分と、検索をおこなう部分からなる。

リンクインデクサ部

リンクインデクサ部は各ページからアンカーを抜き出して、絶対 URI に変換を行ってリンクインデックスを作成する。リンクインデックスにはページ属性のインデックスとページ間属性のインデックスが存在する。ページ属性のインデックスは各ページの正と逆の両方向のリンク先ページを持っている。また、強連結グループも事前に計算して保持されている。ページ間属性のインデックスには、ページ間のリンクの重み (正リンクの数) と、距離、影響度が事前に計算されて保持されている。

属性	内容
links	正リンク先のリスト
rlinks	逆リンク先のリスト
gpages	Web グループページのリスト

表 7.1: ページ属性のインデックス

属性	内容
weight	正リンクの数
distance	距離
influence	影響度

表 7.2: ページ間属性のインデックス

グループエンジン部

グループエンジン部は Namazu からの各ページの得点と、リンクインデックスを元に検索をおこなう。まず、各ページの得点と影響度とから、代表ページを決定する。代表ページ以外のグループ内のページを除去しながら、高得点順に利用者に検索結果を返す。利用者がグループ内のページを結果に含めることも可能である。また、グループにする距離を明示して与えることができる。

7.2 動作例

SunOS 5.7 sparc Ultra-250 の上で動作を確認した。使用した Namazu のバージョンは 1.9.13、Ruby のバージョンは 1.4.3 である。

JAIST を例として、“ネットワーク 研究” をキーとした検索結果に適用した。元となる Namazu による検索結果の上位 10 位の結果は以下の通り。

```
1. Resrach Labs of School of Information Science (score: 214)
http://www.jaist.ac.jp/is/intro/is-labindex.html (10,168 bytes)

2. Lab Homepages of Graduate School of Information Science, JAIST (score: 209)
http://www.jaist.ac.jp/is/intro/is-byname.html (17,592 bytes)

3. Faculty Profiles of School of Information Science (score: 207)
http://www.jaist.ac.jp/is/intro/is-labs.html (15,663 bytes)

4. Center for Research and Investigation of Advanced Science and Technology, JAIST (scor
http://www.jaist.ac.jp/ricenter/index.html (3,616 bytes)
```

5. Kubota Fumito (score: 90)
http://www.jaist.ac.jp/~kouhou/FP/j/is/kubota.html (4,193 bytes)

6. General Information (score: 89)
http://www.jaist.ac.jp/ks/general/concept/information.html (11,120 bytes)

7. Facilities (score: 84)
http://www.jaist.ac.jp/ks/general/facilities/facilities.html (6,452 bytes)

8. Information Links of "Oil disaster of Nakhodka Accidents" (score: 80)
http://www.jaist.ac.jp/misc/nakhodka.html (19,511 bytes)

9. 共同研究成果の紹介 (score: 66)
http://www.jaist.ac.jp/ricenter/gif/JR-1.html (3,516 bytes)

10. ワークステーションの Frontnet 接続の手引 (newscore 65) (score 65)
http://www.jaist.ac.jp/iscenter/join-frontnet.html

Nmazu-Hige による結果は以下の通り。

1. [Unknown Title] (newscore 453) (score 0)
http://www.jaist.ac.jp/is/index-jp.html

2. Center for Research and Investigation of Advanced Science and Technology, JAIST (news
http://www.jaist.ac.jp/ricenter/index.html

3. [Unknown Title] (newscore 146) (score 0)
http://www.jaist.ac.jp/ks/index.html

4. Faculty Profiles of School of Information Science (newscore 120) (score 6)
http://www.jaist.ac.jp/~kouhou/FP/j/is/index.html

5. [Unknown Title] (newscore 107) (score 0)

```
http://www.jaist.ac.jp/iscenter/index-jp.html

6. [Unknown Title] (newscore 84) (score 0)
http://www.jaist.ac.jp/misc/index-jp.html

7. [Unknown Title] (newscore 84) (score 0)
http://www.jaist.ac.jp/open.class/open-class.html

8. Information Links of "Oil disaster of Nakhodka Accidents" (newscore 80) (score 80)
http://www.jaist.ac.jp/misc/nakhodka.html

9. ワークステーションの Frontnet 接続の手引 (newscore 65) (score 65)
http://www.jaist.ac.jp/iscenter/join-frontnet.html

10. Career Information Service (newscore 63) (score 54)
http://www.jaist.ac.jp/jimu/syomu/koubo/index-jp.html
```

newscore が再計算後の新しい得点で、score は Namazu による元の得点である。Namazu の 1-3 位がまとまって Namazu-Hige の 1 位に、同様に 4,9 位がまとまって 2 位に、6,7 位がまとまって 3 位に、5 位とさらに下の結果がまとまって 4 位にといった効果が見られる。また、Namazu-Hige の 5,6 位ももっと低い順位の結果が集まって、高い順位となって表れている。

また、CGI としても動作しており、こちらは IRIX 6.3 上の Web サーバ (apache 1.3.6) を用いて、Namazu 1.3.0.10、ruby 1.4.0 で動作を確認した。

図 7.2は goo による「金沢 観光」検索結果上位 20 位より深さ 1 のページ群 (リンク元 URL 数 185、総 (リンク元+リンク先)URL 数 1550、リンク数 1793) を距離 5 までのグループ として、代表のページをリンクによる影響度で選びだした結果である。

グループを表示してみると、goo では第 31 位の金沢観光協会の「いいね金沢」の Web ページが、距離 3 のグループの多数の金沢市観光協会のページの影響度によって、検索結果のより良いものとして、結果として上位に浮上している。また、goo では第 20 位の Web ページは金沢観光協会の「いいね金沢　観光情報」も高得点となっているのがうかがえる。これらの浮上した Web ページは 5.1 節などのグループを代表するものとして、適切な Web ページとなっている。

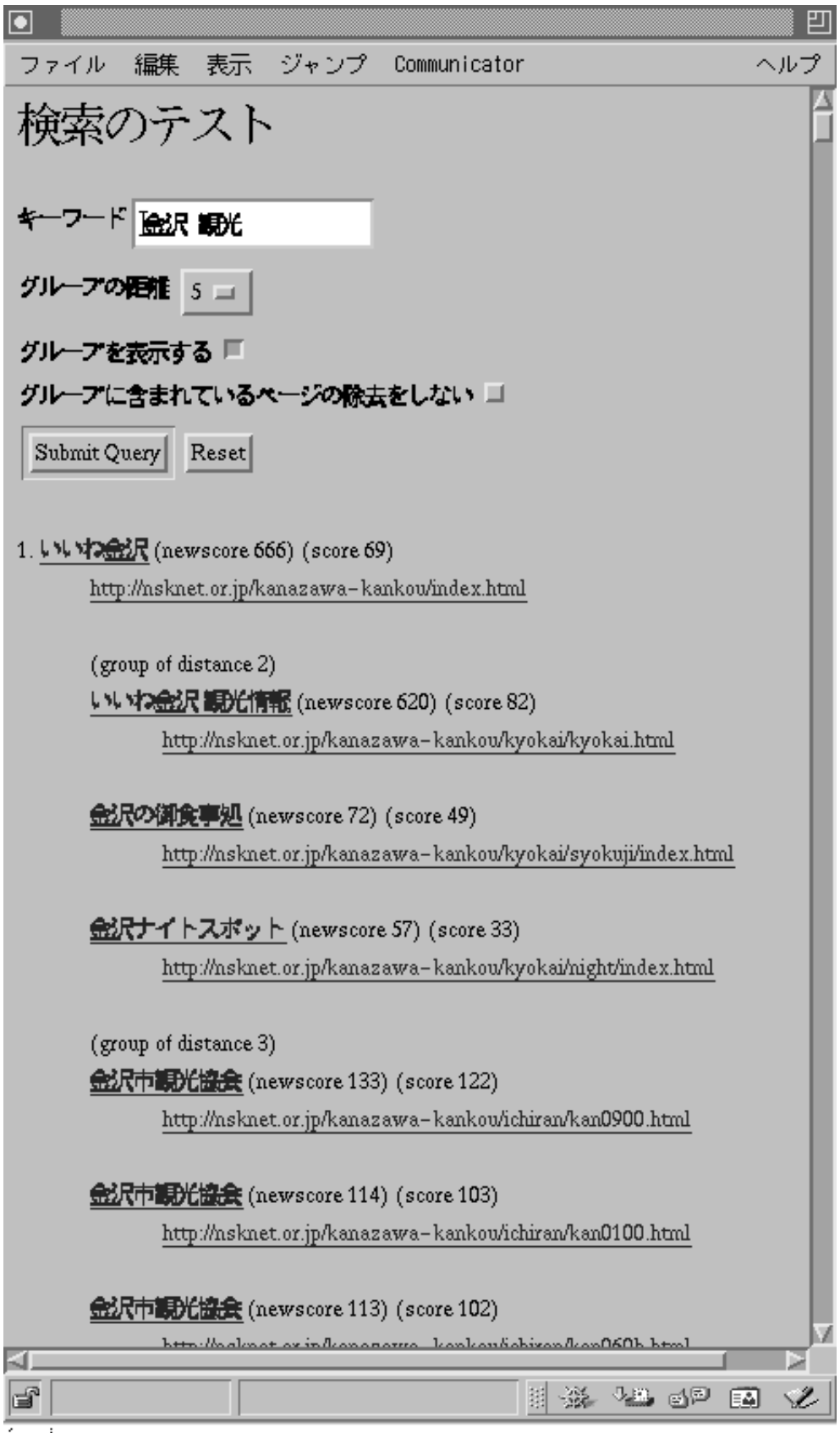

図 7.2: 金沢観光の検索例

7.3 プロトタイプからの考察

本方式は既存の全文情報検索との親和性が高い。例えば、Namazu-Hige で計算された新得点を Namazu のインデックスに反映させたインデックスを新たに生成するようにできる。そうすれば、インデックスのデータ部分だけを差し換えるだけで済み、検索システム自身は既存の Namazu を使い続けるということも可能である。

クレバープロジェクトや googl の研究では、Web 空間全体での再帰的計算が必要で、Web 空間を分類、分割して扱う機能が弱いところの問題を、今後解決する必要がある。本方式の強連結グループの距離 N を無限大にして、被影響度の代わりに影響度を使用し、文章情報を使わない情報検索は、google や clever の情報検索と同じになる。本研究の内容をさらに発展させれば、局所的なリンク構造を取り出して、Web 空間をうまく分割していくことが見込まれる。

本方式は、得点の計算をおこなう範囲を局所に抑えている。それと同時に、文章中から補助として使える情報を最大限に利用している。全体で得点を計算しているために、大局的に信用度の高いグループを導き出すが、局所的に閉じているグループは漏れてしまう。また、彼らの方式は影響度を使っているが、本方式は強連結グループと被影響度を使っている。彼らは影響度でリンク集などのリンクされていない排除しているが、本方式では強連結性でリンク集などが排除されることになる。逆にリンク集などが必要なら、強連結性でグループ化をしなければ、自動的に見つけることができる。

1 つのページから距離 N までの計算量は 1 ページの平均リンク数を L として、$O(N^L)$。各ページで計算するので全体では、ページ数を P として $O(P \times N^L)$。ちなみに、JAIST では L=4.9。1 つのページから距離 5 までのグループの検索に現在は Ultra-250 で数分かかっている。

将来の計算能力の増大に合わせて、距離を伸ばし、より詳細に文章情報を利用するようになれば、彼らの方式をも合わせ込んだ強力な情報検索方式へと発展を遂げるであろう。

従来の検索システムの評価には適合率と再現率が用いられることが多い。一方で本研究では、あえて代表ページ以外の Web グループ内のページを結果から除去している。これは利用者に提供する上位結果の情報量をなるべく増やすためである。(また、いうまでもなく Web グループ内のページは代表ページからたどり着ける。) 結果として全体的には適合率と再現率が低くなる恐れが存在するが、利用者にはより利用しやすい結果が得られる。今後のリンクを用いた検索システムの評価方法に、本方式のように上位の検索結果の適合率と情報量が用いられるように、働きかけていく予定である。

第 8 章

将来の展望

8.1 課題

本研究ではいくつかの有効例による有効性までしか示せていないので、今後に定量的な検証を行なう必要がある。例えば、リンクの分類が正しく分類されているのか統計的に検証し、分類の精度を上げるためにパラメータを調整していく必要がある。また、グループ化、階層化、順序づけのアルゴリズム、計算量などを改善するためにも定量的な評価が利用できる。

検索結果以外へも、本研究の手法を応用できる分野がないかを検討している。bookmark ファイルに適用して、その bookmark ファイルの性質を推定したり、逆に望む性質のリンクを持つコンテンツの製作の助けとなるシステムなどが期待される。

8.2 展望

初期の HTML では、リンクは一定の文字または画像からなるエリア (アンカー) が、他のメディア (の先頭または一部のエントリポイント) への参照を示しているだけであった。その後、フレームセットなどが導入されたりしたが、HTML4.0[13] や XML1.0[14] ではリンクのタイプを指示したり、指示するエリアの指定、双方向リンク、リンク群なども規定されるようになっている。リンクに付加されたこれらの情報を、リンクの構造情報と共に扱うことが出来るようになると、さらに効率よくリンク構造をグループ化、階層化することができ、情報検索がより強力なものとなるであろう。

既存の検索エンジンは企業によって開発されたものが多数を占めているためか、速度が重視されすぎている嫌いがある。例えば一案として、一晩かけてじっくりと検索を行うエ

ンジンがあっても良いであろう。

また、企業の手法はブラックボックス化していて、アルゴリズムやパラメータなどの、情報が公開・共有されにくい点も、検索エンジンの研究の発展の障害の 1 つとなっている。その中で、オープンな研究が増えてきているのは力強いことである。

キーワードだけでは全員の目的に合った検索エンジンは難しく、今後はキーワードではない部分で、どうやってより多くの利用者の目的に合った検索を実行できるようにするかが 1 つ大きなポイントであろう。

情報検索では、利用者側でより動的に結果を把握できる仕組みも求められてくる。例えば、納豆ビュー [24] のように、3D 空間で検索結果を把握できるようになるであろう。統一、洗練されたインターフェースの操作環境を利用者へ提供する必要がある。

5.3節では色を用いた結果の視覚化を行ったが、色以外にも、大きさや、形、そして、3次元表現などと、多様な方法が考えられ、利用者の助けには、どのような方法論で取り組めば役に立つのかを検討する必要がある。

今までより対話性の高いインターフェースで、利用者に積極的にシステムにかかわらせることも、検索の効率を改善される。色やグループ化のパラメータが対話的に割り当てることができれば、利用者の望む評価パラメータを検索結果へ適用しやすく、順序の変動、可視化 (色) による直観的距離、関係把握が容易にできる (例えば、各キーワードに重みを持たせて、連続的に変化が可能なインターフェース。) システムとなるであろう。

リンクの構造情報が、グループ化、階層化、順序づけなどに利用ができるようになれば、情報検索だけでなく、今後は多くの分野に利用することが可能となる。例えば、情報のダイジェスティングや、プリフェッチ、キャッシングにも利用が期待される。さらには、情報の別の視点からの再構成することにも利用できる。

情報のダイジェスティングでは、各 Web ページグループに対して、その目次となるページを相対順序などを利用しつつ、Web ページグループが多数存在する場合には、目次の目次となるページをさらに作成して、情報の集約を行う。目次の見出しの部分は、各 Web ページのタイトルやヘディング部分を取りだしたりして、各 Web ページの先頭部分をダイジェストとして加えることで Web ページグループ全体のダイジェストも自動的に作成ができよう。全体をまとめて各キーワードから逆引を可能にする索引機能は将来も、もっと洗練されたものが Web に備わるようになるであろう。

情報の再構成は、既に存在している Web ページ (群) 間の階層化、順序付けを行うだけでなく、例えば、ある Web ページ群のグループの目次や索引となるページを動的に作り出して、利用者のニーズに応じた誘導マップを作ったりするなどといった、利用者に新しい 2 次情報を提供することである。

第 9 章

おわりに

本研究では、Web 上の情報より、リンク構造情報を取り出し、利用すること有効性を述べた。

続いて、Web 上の情報よりリンク構造情報を取り出す方法についての提案を行ない、その妥当性を検証した。

最後に情報検索におけるリンク構造を利用した、グループ化、階層化、順序付け、と実際の応用例を示し、プロトタイプによる評価を行なって、その応用の有効性を示した。

また、今後の課題を論議して、Web をより利用しやすくするために、リンク構造を用いた展望を示した。

謝辞

篠田陽一先生には忙しい中、御指導していただき、感謝致します。

また、井澤 志充、宇夫 陽次朗、田中 友英、三輪 信介の各氏にもゼミをはじめとして様々な御指導をいただき感謝致します。

その他の篠田研究室・落水研究室の皆様には、様々な有形・無形の援助を数多、受けました。ここに深く感謝致します。

最後に、様々な面で辛抱強く私を支えてくれた、父母と妹に深く感謝して、この論文を締め括ります。

参考文献

[1] Yahoo!,
http://www.yahoo.com/

[2] AltaVista,
http://altavista.com/

[3] Google.com,
http://google.com/

[4] Direct Hit,
http://www.directhit.com/

[5] Ahoy!,
http://ahoy.cs.washington.edu:6060/

[6] Jonathan Shakes, Marc Langheinrich & Oren Etzioni,
DYNAMIC REFERENCE SIFTING: A CASE STUDY IN THE HOMEPAGE DOMAIN,
Proceedings of the Sixth International World Wide Web Conference, pp.189-200, 1997,
http://ahoy.cs.washington.edu:6060/doc/paper.html

[7] Inktomi.com,
http://www.inktomi.com/products/search/clustered.html

[8] Northern Light,
http://www.northernlight.com/

[9] goo,
http://www.goo.ne.jp/

[10] Sergey Brin and Lawrence Page,
The Anatomy of a Large-Scale Hypertextual Web Search Engine, WWW7, 1998.
http://google.stanford.edu/long321.htm

[11] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd,
The PageRank Citation Ranking: Bringing Order to the Web 1998.

[12] 片山研一,
HTML 文書のカテゴリ階層への自動割り当て
北陸先端科学技術大学院大学 情報科学研究科 奥村研究室1998.
http://www.jaist.ac.jp/library/thesis/is-master-1998/paper/k-kataya/paper.ps

[13] HTML 4.0 Specification,
http://www.w3.org/TR/REC-html40/

[14] Extensible Markup Language (XML) 1.0,
http://www.w3.org/TR/REC-xml

[15] NTT,
ーインターネット上に仮想統合データベースを構築可能ー情報検索システム‘ Ingrid（イングリッド）’を開発
NTT R&D Vol.45 No.10 pp.1059
http://www.w3.org/Conferences/WWW4/Papers/300/

[16] Narayanan Shivakumar, Hector Garcia-Molina,
Finding near-replicas of documents on the web
Proceedings of Workshop on Web Databases (WebDB'98) held in conjuntion with EDBT'98 Mar 1998.
http://www-db.stanford.edu/~shiva/Pubs/web.ps

[17] T. Berners-Lee,
Universal Resource Identifiers in WWW
http://www.rfc-editor.org/rfc/rfc1630.txt

[18] T. Berners-Lee, D. Connolly,
Hypertext Markup Language - 2.0
http://www.rfc-editor.org/rfc/rfc1866.txt

[19] R. Fielding, UC Irvine, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee,
Hypertext Transfer Protocol – HTTP/1.1
http://www.rfc-editor.org/rfc/rfc2068.txt

[20] 山名早人,
WWW 情報検索サービスの動向
http://www.etl.go.jp/~yamana/Research/WWW/survey.html

[21] SEARCH ENGINE WATCH,
http://www.searchenginewatch.com/

[22] IBM アルマデン研究所,
クレバープロジェクト
http://www.almaden.ibm.com/cs/k53/clever.html

[23] M. Sarkar and M. H. Brown,
Graphical fisheye views
Communications of the ACM, 37(12) pp 73-84, 1994

[24] 塩澤秀和,
納豆ビュー
http://www.myo.inst.keio.ac.jp/NattoView/

[25] Rong Yang,
Relationship between Precision and Recall ratios and other evaluation methods
http://www.staff.uiuc.edu/~pare/rong.html

[26] Michael Chen, Marti Hearst, Jason Hong, James Lin,
Cha-Cha: A System for Organizing Intranet Search Results
The Proceedings of the 2nd USENIX Symposium on Internet Technologies and SYSTEMS (USITS), October 1999.

[27] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins.
Hypersearching the web.
Scientific American, June, 1999.

[28] Namazu,
http://openlab.ring.gr.jp/namazu/

[29] Ruby,
http://www.ruby-lang.org/

[30] Namazu-Hige,
http://shinoda-www.jaist.ac.jp/Projects/hige/

[31] 馬場肇,
日本語全文検索システムの構築と活用
ソフトバンク、 1998 年 9 月

[32] リチャード・セルツァー, エリック・J・レイ, デボラ・S・レイ,
ALTAVISTA 完全活用ガイド
翔泳社, 1997 年 11 月

付 録 A

プロトタイプのプログラムの入手方法

プロトタイプは Web 経由で以下の所より入手可能である。

http://shinoda-www.jaist.ac.jp/Projects/hige/

また、Namazu のインデックスに、リンクの影響を反映させるシステムを実装して、Namazu に還元する予定である。