筑波大学大学院博士課程

システム情報工学研究科修士論文

情報指向型検索のための
情報収集支援インタフェース

小林拓海

(コンピュータサイエンス専攻)

指導教員　田中二郎

2007年3月

概　要

インターネットを用いた情報指向型検索とは、特定のトピックに関する総合的な情報を得るために必要な Web ページ群を見つけるための検索である。現在の Web 検索を用いて情報指向型検索を行う際、ユーザは膨大な Web 検索結果を巡回し、複数の Web ページから必要な情報を探索する必要がある。本研究では、現在の Web 検索の現状と問題点について考察し、既存の Web 検索インタフェースを用いて情報指向型検索を行う場合に発生する問題点を解決するためのインタフェースを提案し、試作システムの実装を行った。

本稿ではまず概観提示インタフェースについて述べる。概観提示インタフェースは Web 検索結果を分析し、内容に応じて分類した検索結果を 1 画面に納めてユーザに提示する。概観提示インタフェースを用いることでユーザは Web 検索結果の全体像を理解することが容易になり、必要な情報を効率よく取捨選択することが可能となる。

また、既存検索インタフェースにおける情報の保存と利用を行う際の問題点に着目し、それらを解決するための機能の拡張を行った。具体的には、概観提示インタフェースに用いている Hyperbolic Tree に Web ページの要旨を付加情報として与える。また、必要な情報を部分的に保存してスクラップブックを作成し、情報指向型検索を行って収集した情報を利用するための機能を提供する。概観提示インタフェースの検索結果提示機能と情報保存機能を組み合わせることで、情報検索と情報の保存を一連の作業として行うことが可能となり、特定のトピックに関する情報収集やレポートの作成など、情報指向型検索を行う具体的な場面で本インタフェースを利用することが可能となる。

目 次

図 目 次

表 目 次

第1章　はじめに

現在、インターネットはより身近なものとなり、情報を収集する場面において欠かせない存在となっている。インターネットを用いる事で我々は世界中のあらゆる情報に容易に触れることができるようになった。インターネットの情報量及び Web ページの数は急速に増加し続けている。例としてサーチエンジン Google[4] は 2005 年現在約 80 億もの Web ページから検索を行っている[1]。また Internet Systems Consortium 社が年に 2 回行っているインターネットホスト数調査の調査結果 [1] によると、2006 年 1 月現在のインターネットに接続されたホストの数は約 3 億 9 千万ホストにも及ぶ。図 1.1は過去 10 年間のインターネットに接続されたホスト数の推移を表したグラフである。インターネットの情報量が急速に増えていることはこのグラフからも明白である。

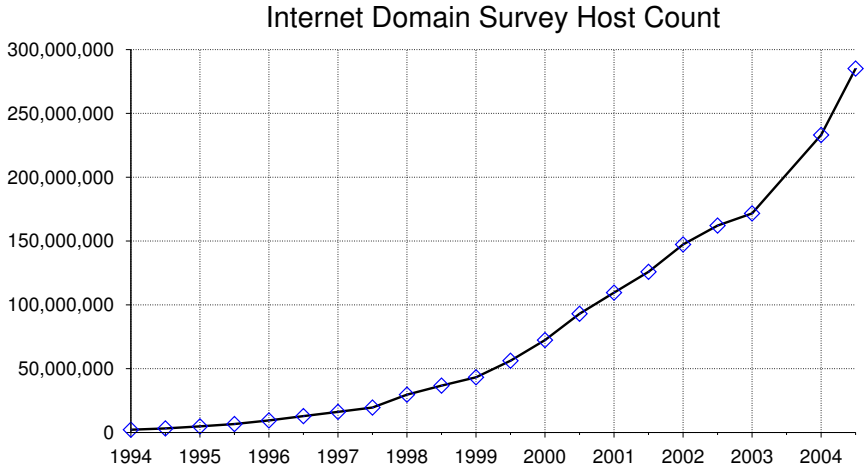

図 1.1: インターネットに接続されているホスト数の推移 (Internet Systems Consortium)[1]

このように膨大な数の Web ページの中からサーチエンジンを用いて検索を行った場合、必然的に検索結果として提示される Web ページの数も膨大になることが多い。そのため、利用者にとって有益な情報を取捨選択することや、Web 検索結果の大まかな全体像を直感的に理解することはますます困難になっていくであろうと予想される。

図 1.2のように、東京に関する様々なジャンルの情報 (政治、交通、観光地、歴史など) を収

[1]現在 Google は検索可能な Web ページを示すインデックス数を公表していない。

集するための検索のように、特定のトピックに関する総合的な情報を獲得するために必要なWeb ページ群を収集するための検索を情報指向型検索と呼ぶ。本研究では Web 検索の現状を踏まえ、情報指向型検索を行う際の既存インタフェースの問題点を解決するインタフェースについて述べる。

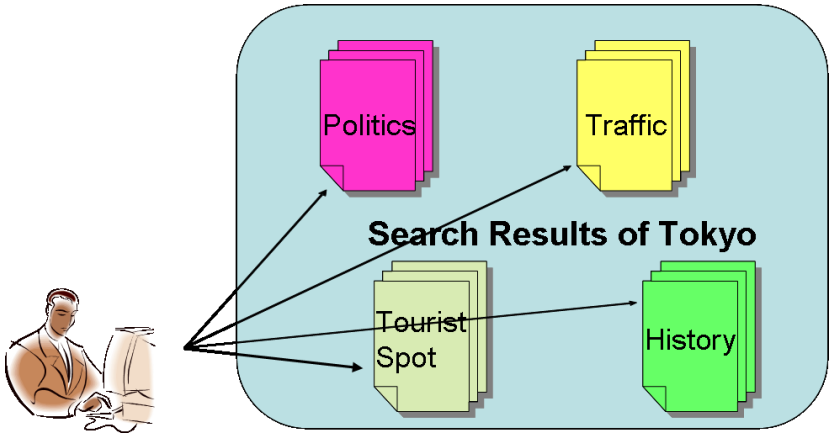

図 1.2: 情報指向型検索

本研究の目的

本研究の目的は Web 検索の現状を理解した上で既存の Web 検索インタフェースにおける情報指向型検索の問題点を解決するインタフェースを考案し、実装することである。

情報指向型検索における検索結果の特徴に対する直観的理解と情報収集を支援するために本研究では Web 検索結果として得られた複数の Web ページに対する内容分析および分類を行い、分類結果を 1 画面に納めてユーザに提示する概観提示インタフェースの提案と試作を行った。

さらに情報指向型検索における情報の収集・保存および情報の利用をより効率よく行うための機能拡張を行う。概観提示インタフェースに本研究で提案する機能拡張を行うことで、情報検索と情報の保存を一連の作業として行うことが可能となり、具体的な情報指向型検索のタスクでの利用が可能となる。

本論文の構成

本論文の構成を述べる。第 2 章では我々が日常的に行っている Web 検索および既存検索インタフェースについて考察し、情報指向型検索を行う際の問題点を述べ、関連研究をあげる。第 3 章では問題解決手法として概観提示インタフェースを提案し、その特徴と利用する要素技術について説明し、さらに実装と詳細について述べる。、第 4 章では概観提示インタフェー

スの機能拡張について述べる。第 5 章では本インタフェースの利用シーンの例を挙げ、第 6 章でインタフェースの考察を行い、最後に第 6 章でまとめる。

第2章　Web検索の現状と問題点

本章ではまず日常のWeb検索の現状を考察する。さらに既存検索インタフェースを利用して情報指向型検索を行った場合の問題点について述べ、本研究の目的をより明らかにする。

2.1　Web検索の現状と考察

IBMのAndrei Broderは、Web検索におけるユーザが与えるクエリの背景にある情報ニーズを次の3つのカテゴリーに分類している[5]。

1. 情報指向　(informational)
2. ナビゲーション指向　(navigational)
3. トランザクション指向　(transactional)

1の情報指向型の検索とは、特定のトピックに関する1件もしくは複数件のWebページを獲得することを要求する検索である。例えば「つくば」に関する様々な情報を収集したい場合などはこの検索に当たる。

2のナビゲーション指向型の検索とは、ある特定のWebサイト(またはある対象物の代表的なページ)に到達することを要求する検索である。例えば筑波大学のホームページに到達することを目的とするような検索はこれに当たる。

3のトランザクション指向型の検索とは、インタラクションを伴うようなWebサイト(オンラインショッピング、Webが仲介する様々なサービスなど)に到達することを要求する検索である。例えばつくばにあるホテルから自分の要求に合ったホテルを探し、予約することを目的としたような検索はこれに当たる。

我々は以上のような種類のWeb検索を日常的に行い、必要な情報を得ているということができる。ナビゲーション指向型の検索は具体的な特定のWebサイトに到達することを目的としているため、一般的な検索サイトからでもたどり着くことは比較的容易である。例えば検索サイトGoogleは、ナビゲーション指向型の検索に優れており、ユーザが目的とするWebページを上位に表示するための様々な工夫がなされている。またGoogleはナビゲーション指向型検索に特化した機能を備えており、クエリを入力し“I’m Feeling Lucky”ボタンをクリックするとほとんどの場合目的のWebページのトップページへ到達することが可能になっている。

現在日本で最も多くのユーザに利用されていると考えられる検索サイトはディレクトリ型検索を行うYahoo!、またはロボット型検索を行うGoogleである。また、infoseek、goo、Excite、

Biglobe、@nifty、So-net、AOL、DION などの大手検索サイトのほとんどが Google のシステムを共有し、その結果を反映している。これらの検索サイトの多くは、ユーザがクエリを入力し、検索エンジンが検索した多くの Web ページをテキストのリストで提示するというインタフェースになっている。ここで言うテキストのリストとは Web ページのタイトル、クエリを含む文章の抜粋などである。例えば Google を用いて「つくば」というクエリで検索を行った場合、ユーザには図 2.1のような画面が表示される。この場合の検索結果は約 125 万件という膨大な数であり、Google など多くの検索サイトでは検索結果の中から 1000 件のみを数十ページに分割して提示している[1]。

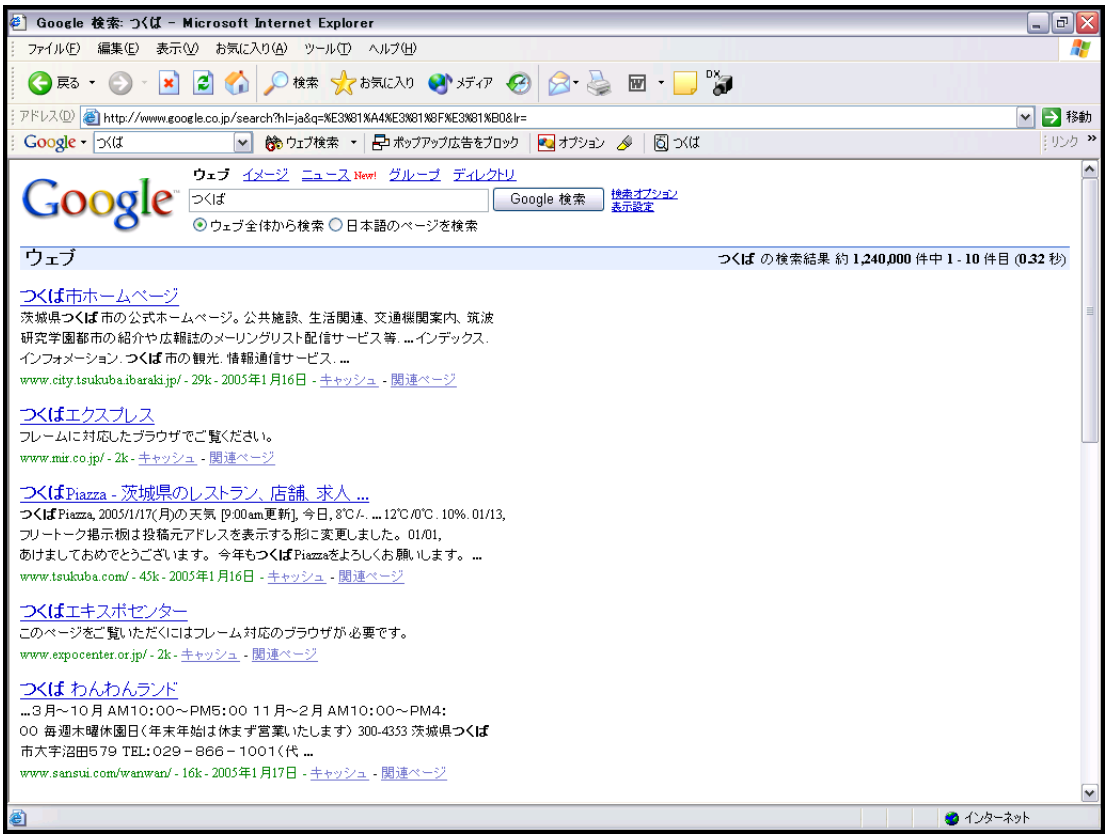

図 2.1: Google の検索結果提示画面

一般に Web 検索で用いられるクエリは短く、利用者はランキング検索結果の上位しか閲覧しない傾向がある。英語圏の Web サーチエンジン Excite の場合、検索質問の長さは平均 2.21 語であり、利用者は 1 ページに 10 文書ずつ表示される検索結果を平均 2.35 ページしか閲覧していない [6]。日本語では複合語が用いられるため，検索語数はより小さくなる。日本語 Web ページを主な検索対象とする Web サーチエンジン ODIN[2]において、1ヶ月に用いられた検索

[1]Google では検索用語に最も関連性のある検索結果が 1000 件以上ある場合でも 1000 件のみを表示する。
[2]http://odin.ingrid.org/　　現在はサービスを休止している。

質問に含まれるクエリは平均 1.40 単語であり、検索質問の 7 割以上が 1 つのクエリのみから構成されるという事実もある [7]。検索質問のクエリの数が少なければ、絞込みの条件が少なくなるために、検索結果は膨大な数となることは明らかである。また、数十ページに渡る検索結果がありながらユーザはごく一部のみしか閲覧していないということは、ユーザにとって有益な情報を見落としている可能性があることを示している。

2.2 情報指向型 Web 検索における既存インタフェースの問題

2.2.1 Web 検索結果の提示法に関する問題

情報指向型の検索を行う場合、ユーザは膨大な検索結果を様々な観点から眺め、必要な情報を取捨選択する必要がある。前節で例を挙げた「つくばに関する様々な情報を収集する」という目的の検索を行う場合を考えてみる。

「つくばに関する情報」は抽象的であり、市政に関する情報、交通に関する情報、ショッピングに関する情報、宿泊施設に関する情報など様々である。そのためユーザは特定の Web ページのみから情報を得るのではなく、様々な Web ページを巡回しながら情報を収集することになる。この場合 Google のような検索結果提示インタフェースでは様々なジャンルの情報が混在しているため、ユーザの情報収集の能率を阻害していると考えられる。あるジャンルの情報を収集する際にはジャンルごとに Web ページがまとまって提示されていたほうが情報を収集しやすい。

また、検索結果が数十ページにわたって分割されて提示される表示法も、検索結果全体の概観の理解ができず、有益な情報を見落としてしまう場合がある。また、このような表示法ではランキングの後方にあるページはほとんど閲覧されないため、有益な情報がある可能性があるにもかかわらず情報が発見される前に検索そのものを打ち切ってしまうなどといった問題点が考えられる。検索結果が 1 画面に納まっていて特徴を表す概観が直観的に理解できたほうが情報の見落としが防げる。

膨大な数の Web 検索結果を適切にジャンルごとに分類し、それらを 1 画面に納めてユーザに提示することができれば、これらの問題点を解決し、ユーザが行う情報指向型検索を支援することができる。

2.2.2 情報の収集と保存・情報の利用を行う際の問題

既存検索インタフェースを用いて Web 検索を行う場合、ユーザはシステムが提示した検索クエリを含む Web ページの抜粋などを参考に検索結果提示画面と実際の Web ページ間を行き来する。複数の Web ページを巡回して情報を収集する必要がある情報指向型検索を行う際にはページ間の行き来による複数回の画面の切り替えがユーザの情報収集効率を悪くしている。この問題を解決するためには実際の Web ページの閲覧をなるべくすることなくユーザが必要な情報が含まれるページか否かを判断するための付加情報をユーザに適切に提示する必要が

あると考えられる。

また、必要な情報を発見した際にはWebページ全体を保存したり、必要な箇所をメモするなど、情報検索と情報の保存が一連の作業として行われないという問題がある。本インタフェースでは、情報指向型検索における検索機能と情報保存機能を組み合わせることで、情報検索と情報の保存を一連の作業として行うための手法を提供し、ある検索クエリに対するスクラップブックを作成することができる。ユーザは情報指向型検索で収集した情報を元にスクラップブックを作成し、必要な際にそれを閲覧することでレポート作成などの具体的な場面で本インタフェースを利用することが可能となる。

2.3 関連研究

Poznan University of Economics, Department of Information Technology の開発する Periscope[8] は本研究と同様に Web 検索結果の概観をユーザに提示するインタフェースとなっている。ユーザは 3D 空間に Web ページを表す 3D オブジェクトが配置された画面を提示される。Web ページの分類はページのタイプ、ホスト名、使用言語、サイズなどで分類されている。本研究とは 2D 空間に概観を表示している点、またクラスタリングの際 Web ページの内容に着目している点で異なっている。Web ページの内容を考慮しないクラスタリングではユーザへの情報収集支援が十分とはいえない

University of Kent at Canterbury の Jonathan Roberts らは Web ページを分類した結果を二次元空間に表現した [9]。分類にはホスト名を用いている。さらにクラスタの配置には、Web ページのインリンクとアウトリンクの数をそれぞれ x 軸、y 軸に対応させている。本研究は Web 文書を内容でクラスタリングし内容の近いページほど近くに配置する。Web ページの配置に関してリンク構造に着目している点が本研究と異なる。

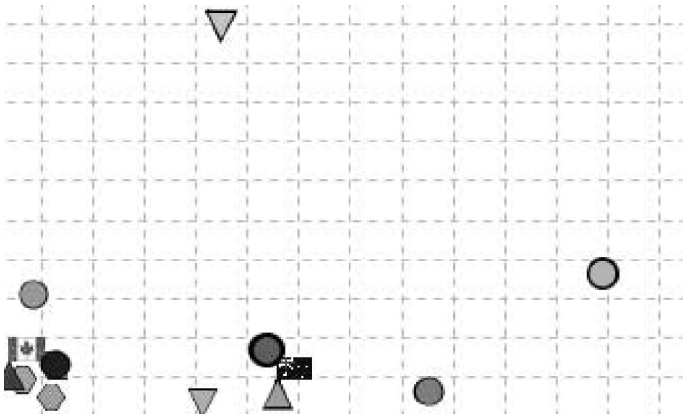

図 2.2: Web ページをホスト名でクラスタリングし二次元空間に配置した例

Palo Alto Research Center,Information Sciences and Technologies Laboratory の J.D.Mackinlay らは長いリストを三次元空間上で曲がった壁に貼り付けるインタフェース Perspective Wall[10] を提案した。Perspective Wall は中央の壁にあるリストは大きく表示され、左右の壁にあるリストは小さく表示される。一次元のリストを表示する場合には有効だが、本研究のように二次元の樹形図を表示する場合には適していないと考えられる。

Web 上で自由に使用することができる関連インタフェースとしては、Vivisim[11]、Clusty[12]、KartOO[13] などがある。Vivisimo(図 2.3) や Clusty は、Web 検索結果を動的にユーザに提示するインタフェースである。これらのインタフェースは、検索クエリを与えられると画面左にクラスタリング結果、画面右に検索された Web ページがそれぞれ一次元のリストで提示される。これに対し、本研究の概観提示インタフェースはクラスタリング結果を Hyperbolic Tree を用いて視覚的にユーザに提示するため、ユーザは検索結果の全体像をより直感的に理解す

ることが可能である。

図 2.3: Vivisimo のクラスタリング結果提示画面

また、KartOO(図 2.4) はメタ検索エンジンであり、検索結果を Flash を用いたマップで表示する。検索結果の Web ページの「近さ」が地図の相関関係で分かるようになっている。KartOO では本研究と同様に関連のある Web ページ同士が関連を表現するラベルによって接続されているが、Web ページの内容を表現する要旨が存在しない、一度に提示できる検索結果が少ないなどの問題点があると考えられる。

図 2.4: KartOO

第3章　概観提示インタフェース

本章ではまず情報指向型検索を行う際の既存インタフェースの検索結果提示法の問題点を改善する手法として概観提示インタフェースを提案する [14][15]。次に提案インタフェースに関連した要素技術を各々簡単に説明し、概観提示インタフェースの実装の詳細を具体的に述べる。

3.1　概観提示インタフェースのための提案

本研究では、前章に述べた情報指向型検索における既存インタフェースの提示法に関する問題点を改善する方法として、概観提示インタフェースを提案する。概観提示インタフェースは以下の要件を満たすものとする。

要件 1 : 類似ページを二次元空間において近傍に配置

要件 2 : 検索結果を一画面に納めてユーザに提示

要件 1 を満たし、検索結果を二次元空間を用いて提示することで、リストを順に見ていく必要がある一次元による提示インタフェースよりも提示表現の幅を広げることが可能となり、ユーザはより柔軟に Web 検索を行なうことができるようになる。また、類似ページを近傍に配置するために検索結果の Web ページをページの内容によってクラスタリングする。これにより様々なジャンルのページをあちこちに散在させることなく近傍に配置することで、ユーザはより効率よく情報を収集することができる。さらに、クラスタリングを行なうことにより、検索結果に含まれるユーザの意図しない Web ページもジャンルごとに近傍に配置されることになる。このため、意図しない Web ページの散在による情報収集の際の弊害を取り除くことができる。

要件 2 を満たすことで分類された検索結果は一画面に納まってユーザに提示される。このため検索結果が数十ページに分割されるインタフェースよりもユーザの検索結果の全体像理解が容易になる。また、情報を提示する際に、本システムでは分類した Web ページをそのまま提示するのではなく、分類結果にラベルを付加して提示する。ラベルはクラスタの特徴を表す単語で構成されている。ユーザはラベルを参考に必要な情報が存在すると考えられる複数の Web ページを容易に発見することができる。

概観提示インタフェースではユーザへの検索結果の提示に “Hyperbolic Tree” を用いた。“Hyperbolic Tree” とは、深い階層構造を持った樹形図を効率よく表現することができる表示方法である。“Hyperbolic Tree” については後に詳しく説明を述べる。

このような機能を実装したシステムを利用することで、検索結果全体としてどのような特徴があるかというような概観の直感的理解を支援することが可能となり、クラスタリングの結果、内容の近い Web ページ同士は近距離に配置されることとなるので、ユーザの情報収集の効率を向上させることができるのである。

3.2 提案手法と他の類似インタフェースとの相違点

ここでは本研究で提案する手法と他の類似インタフェースとの相違点について述べることで本手法の特徴をさらに明らかにする。他の類似インタフェースとしては、概観提示インタフェースと同様に検索結果をジャンルごとに分けてユーザに提示するディレクトリ型検索エンジンを取り上げる。

ディレクトリ型検索エンジンには様々あるが、登録されている Web ページを人手によって決められたカテゴリに分類し、検索結果提示に反映させるという方式のものがほとんどである。図 3.1に検索クエリに「つくば」としてディレクトリ型検索エンジンを用いて検索を行った検索結果提示インタフェースを示した。

図 3.1: ディレクトリ型検索エンジンの検索結果提示画面 (Yahoo!Japan[2])

「つくば」で検索した場合、784 件の登録サイトを 24 のカテゴリに分類した結果を表示している。これは前章に例としてあげたロボット型検索を行う Google に比べて極端に少ない。

これは Google がロボットを用いて Web 上を巡回して自動的に Web ページを収集するのに対して、ディレクトリ型検索エンジンは人手で Web ページを収集しているためである。

表示インタフェースについて見てみると、カテゴリも登録 Web サイトも複数ページにまたがっている。さらにタイトルやページの簡単な説明などの文字の羅列になっており、検索結果全体の概観が直感的に理解できるとは言い難い。

ここで提案手法とディレクトリ型検索との相違点を表 3.1にまとめてみる。まず分類の方法

表 3.1: 提案手法とディレクトリ型検索との相違点

	ディレクトリ検索	提案手法
分類のタイミング	静的	動的
分類の方法	人の手で分類、多くの人手が必要	アルゴリズムによって分類、ほぼ自動的に計算機が分類してくれる
分類の正確さ	正確だが、分類を行う人間の主観が伴う	アルゴリズムによる、ある程度正確
対象の規模	対象文書は少ない	対象文書は多い
検索時の必要時間	時間がかからない	時間がかかる (前処理によって改善可能)
カテゴリ	カテゴリはあらかじめ人手で設定されている	検索キーワードによって同一の Web ページでも属するカテゴリが異なる
検索結果	複数ページにまたがる	1 画面内に納まる

であるが、ディレクトリ型検索ではあらかじめ Web ページはカテゴリに分類されているのに対し、提案手法では検索時に動的に分類を行う。また、ディレクトリ型検索は正確である反面、分類を人の手で行うために多くの人手が必要となるという問題点があるが、提案手法ではアルゴリズムによって計算機が自動的に分類を行う。またディレクトリ型検索は、分類に人間の主観が伴う場合がある。対象の規模は登録された Web ページからのみ検索結果を提示するディレクトリ型検索に比べて、提案手法ではロボット検索で用いられる検索結果を扱うために対象文書は非常に多い。検索に要する時間については、あらかじめ分類処理がされてあるディレクトリ型検索の方が時間がかからないが、提案手法においては Web ページのデータをあらかじめ収集し、分析しておくことで処理時間を短縮できると考えている。各 Web ページが属するカテゴリについては、あらかじめ属するカテゴリが決まっているディレクトリ型検索とは違い、提案手法では検索キーワードによって同一の Web ページでも属するカテゴリが異なる。検索結果はディレクトリ型検索は複数ページにまたがるのに対し、提案手法では 1 画面に納める事が可能となっている。

3.3　概観提示インタフェースで利用する要素技術

本節では Web ページを分類するためのクラスタリング手法や、概観を提示するための可視化法について説明する。

3.3.1　クラスタリング

ここではクラスタリングについて説明し、クラスタリングに用いるベクトル空間法と tf/idf 法についても述べる。

クラスタリングとは

クラスタリングとは、異なる性質のもの同士が混在している集合の中から互いに類似したものを集めてクラスタを作り、集合を分類するための方法である [16]。クラスタリングには様々な種類が存在しているが、本研究では「階層的クラスタリング」を用いてクラスタリングを行った。この手法は対象間の類似度[1]を手がかりにして、樹形図を構成することを目的とするものである。階層的クラスタリングの手法は以下の通りである。

1. 1 つずつの対象を構成単位とする n 個のクラスタから出発する
2. クラスタ間の類似度行列を分析し、最も類似度の高い 2 つのクラスタを融合して 1 つのクラスタを作る
3. 新しく作られたクラスタと、他のクラスタとの類似度を計算して、類似度行列を更新する
4. クラスタが 1 つになっていれば終了し、そうでなければ 2〜3 を繰り返す

本研究において n 個の Web ページのクラスタリングする場合、各 Web ページをクラスタとした n 個のクラスタから出発することとなる。また、ある Web 文書 A とある Web 文書 B の類似度を求める場合には、文書 A の特徴と文書 B の特徴をそれぞれベクトル空間法で表現し、類似度を求めることになる。

形態素解析

文書の特徴を分析するにはその文書にどのような単語が出現するかを調べる必要がある。そのために用いるのが形態素解析という技術である [17]。形態素解析とは、文字列を単語の列に分割し、それぞれに品詞や語形変化の情報を与える処理のことである。表 3.2は「私は筑波大学に通っています。」という文章に形態素解析を行った例である。

[1] ここでいう類似度とは値が大きいほど類似性が高いということを示す数値である

表 3.2: 「私は筑波大学に通っています。」を形態素解析した例

私	ワタシ	私	名詞-代名詞-一般	
は	ハ	は	助詞-係助詞	
筑波大学	ツクバダイガク	筑波大学	名詞-固有名詞-組織	
に	ニ	に	助詞-格助詞-一般	
通っ	トオッ	通る	動詞-自立	五段・ラ行 連用タ接続
て	テ	て	助詞-接続助詞	
い	イ	いる	動詞-非自立	一段 連用形
ます	マス	ます	助動詞 特殊・マス	基本形
。	。	。	記号-句点	

このように形態素解析を行って得られた単語から、不要語を除いた単語の出現頻度を計測したものを、文書の特徴語とし、類似度判定に用いた。不要語とは助詞や助動詞、記号など単語のみでは意味を成さない語である。なお、本研究においては検索クエリも不要語に含んでいる。検索クエリは検索結果の Web ページすべてに存在し、特徴を表す単語にはなりえないと考えたためである。

tf/idf 法

形態素解析を用いる事で、1 つの Web 文書の中の特徴語の出現頻度は求めることができるが、Web 検索結果全体に対して文書中の単語がどれほど重要であるかは分からない。そこで用いられる手法が tf/idf という手法である。

tf/idf 法を用いることによって、「ある単語の、その文書における文書集合全体を考慮した相対的な重要度」を算出することができる。文書 D_i の中の単語 t_j の重み (重要度)w_{ij} を求める場合以下の計算式となる。

$$w_{ij} = tf_{ij} \times idf_j \tag{3.1}$$

tf_{ij} とは、局所的重みとも呼ばれ文書 D_i の中での単語 t_j の出現率を表現している。文書 D_i に単語 t_j が多く出現すればするほど、tf_{ij} は大きな値となる。

idf_j とは、大域的重みとも呼ばれ、単語 t_j が全文書集合の中に出現すればするほど idf_j は小さな値となり、珍しい単語であれば大きな値となる。

まとめると、ある文書 D_i における単語 t_j の重要度 (重み) は、文書 D_i において単語 t_j の出現頻度が高く、かつ全文書集合中の出現頻度が小さい場合に大きくなるといえる。tf/idf の計算法については、様々なものが考えられるが、本研究で用いた計算式については、次章で述べる。

ベクトル空間法

ベクトル空間法とは、文書やクエリの内容を多次元空間上のベクトルとして表現する手法である。これには tf/idf を用いて得た重みを適用する。m を文書集合全体の単語数、w_{ij} を文書 Di 中の単語 t_j の重みとすると、文書 D_i はベクトル d_i で表現される。

$$d_i = \left[\begin{array}{cccc} w_{i1} & w_{i2} & w_{i3} & \cdots & w_{im} \end{array} \right] \tag{3.2}$$

このようなベクトルを検索結果の Web ページの数だけ計算し、階層的クラスタリングの説明時に述べた行列計算を行うこととなる。

3.3.2 クラスタリング結果の可視化法

focus+context

表示する画面の制限や人間の認知能力の制限から、一度に表示すべきデータ量は制限される。この制限のもとに、ユーザの知りたいことをユーザの観点でわかりやすく表示する手法として focus+context とよばれる技術が研究されている。

巨大な木構造を画面に表示する場合、一度に全てを表示すると各ノードが小さくなりすぎてしまう。そこで通常の手法では画面の拡大・縮小とウインドウのスクロールを利用する。しかしこの手法では，拡大画面に表示される部分 (注目点:focus) が全体の中でどのような位置をしめるか (コンテクスト:context) がわかりにくい。focus+context 手法は注目する点 (focus) を詳しく表示するとともにそれ以外の部分の概観 (context) を表示する技術である。

focus+context 手法の一般的な枠組の研究例では、Furnas が行った Generalized Fisheye View[18] が先駆的である。この手法では、木構造の各ノードが持つ意味的な重要度と現在のユーザの視点からそのノードまでの距離をもとに、そのノードの表示上の重要度を決定する。意味的に重要なノードでも現在の視点から遠ければその分だけ表示上の重要度は低くなる。この値に基づき、ある閾値以下の重要度のノードは画面上から消去される。この閾値を上下させることで表示されるノードの数を調節することができる。さらに，Sarkar らは Fisheye View の考えを発展させ、グラフ描画の際のノードの配置手法に応用した [19]。この手法は、重要度の高いノード (とその近傍) を大きく、重要度の低いノードを小さく表示する一般的な枠組を定式化している。

Hyperbolic Tree

クラスタリング結果の可視化は Hyperbolic Tree を用いて行った。Hyperbolic Tree とは、John Lamping[3] らによって提唱された、双曲空間上に樹形図を表現する focus+context 手法である。Hyperbolic Tree を用いる事で、深い階層構造を持った樹形図を効率よく 1 画面に納めることが可能となる。Hyperbolic Tree の特徴は、中央に近い部分木ほど大きく表示され、中央から離れた部分木ほど小さく表示される。Hyperbolic Tree の例を図 3.2に示す。実装上ではマウスドラッグでフォーカスを移動させることができる。フォーカスを移動させることで中央から

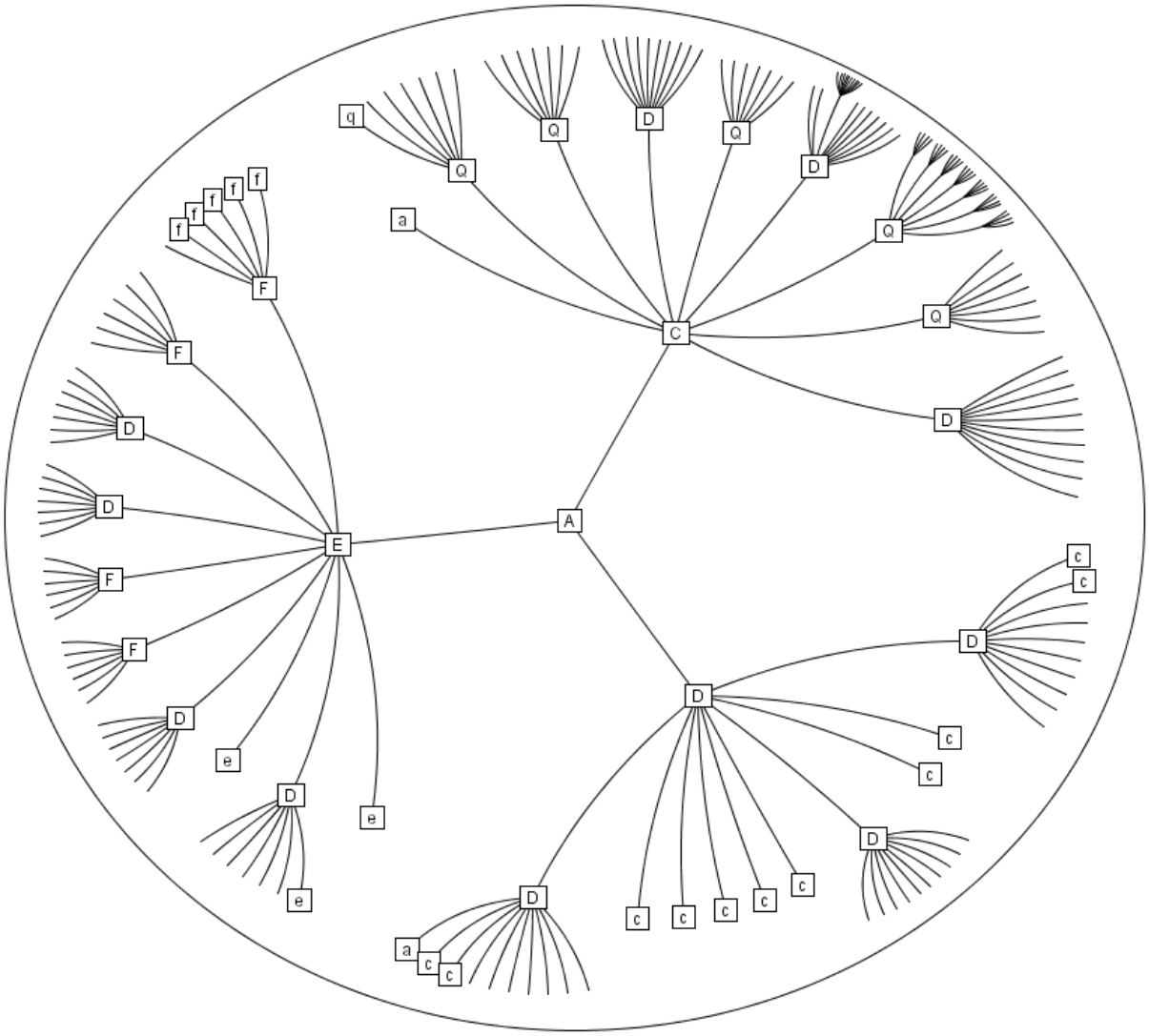

図 3.2: Hyperbolic Tree(John Lamping)[3]

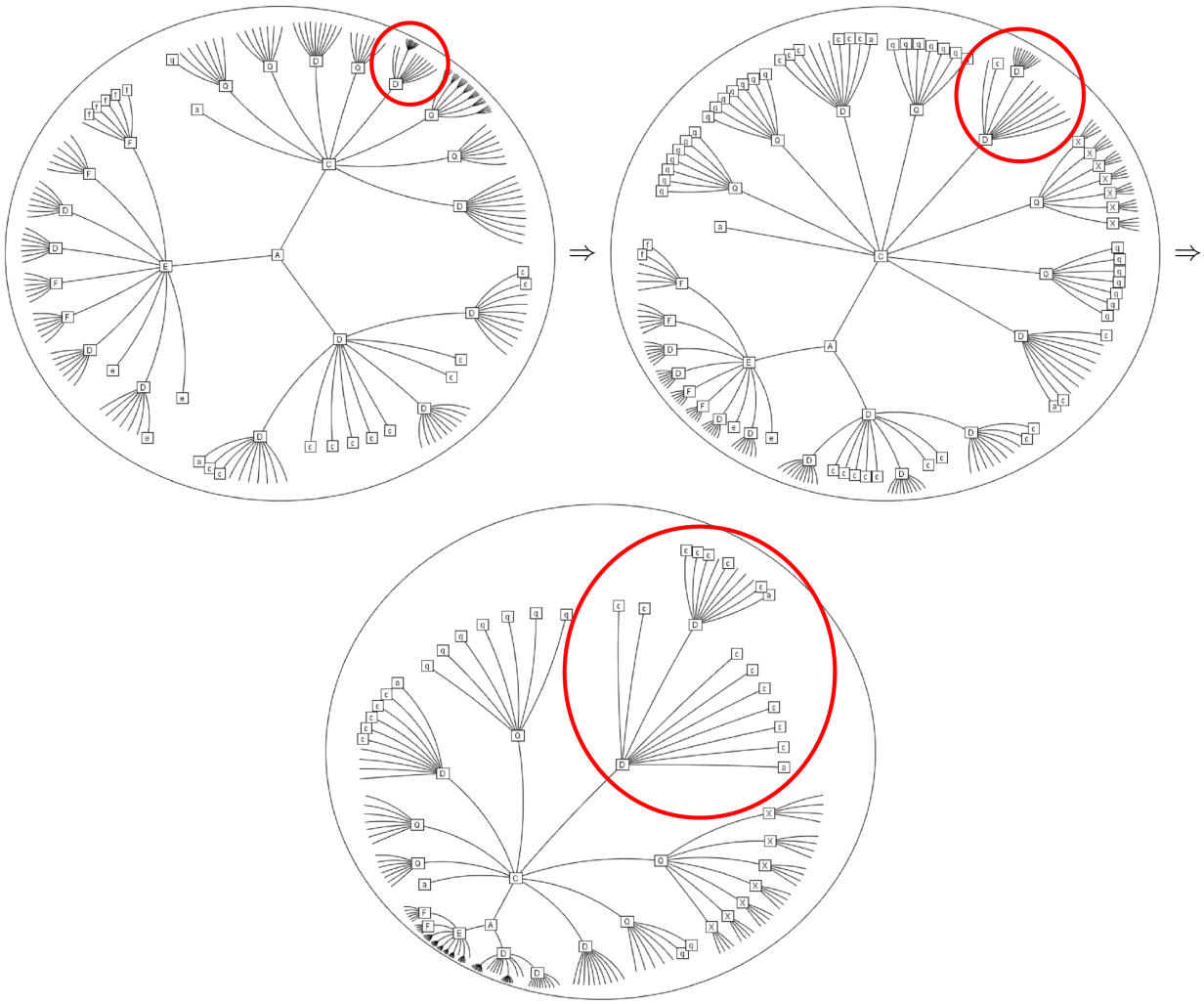

図 3.3: Hyperbolic Tree のフォーカスの移動

離れた部分木が中央に近づき、小さく表示されていた部分木が大きくなる。このような仕組みのため、通常の樹形図よりも 1 画面に多くの情報を取り入れることが可能となるのである。図 3.3にフォーカスの移動の様子を示す。右上の部分木を中央に移動させることでフォーカスが移動し、より多くの情報がユーザに提示されるようになる。ユーザはこのようにして必要な情報にフォーカスを移動させることで、概観を保ったまま必要な情報に注目することができる。

3.4 概観提示インタフェースの実装

ここでは、概観提示インタフェースの実装についての詳細な説明を検索結果のクラスタリング部と概観提示部の 2 つの部分に分けて述べる。

3.4.1 検索結果のクラスタリング部

検索質問の入力と検索結果の取得

ユーザはまずシステムに対して検索クエリを与える。検索クエリは 1 語以上の単語である。するとシステムは GoogleAPI[20] を利用して検索結果を得る。ここで言う検索結果とは、検索クエリに対応する Web ページの URL を意味している。

HTML ファイルへの変換

次に検索結果として得られた URL から個々の Web ページを分析する。まず検索結果の URL から HTML ファイルを得る。この処理には Perl モジュールである HTTP::Lite を用いた。

HTML ファイルの分析

HTML ファイルの分析には、前章で説明した要素技術である形態素解析や tf/idf 法などを用いる。

HTML ファイルは単純な文章ではなく、タグと呼ばれるコマンドを用いて木構造的に構成されている。この木構造を構文解析し、タグの情報を積極的に利用することで、Web ページの特徴をより顕著に抽出できるのではないかと考えた。表 3.3に HTML 文書に現れるタグの一例を挙げた。

表 3.3: HTML 文書におけるタグの例

META	文書情報の記述	TITLE	Web ページのタイトル
CENTER	中央寄せ	STRONG	強調
A HREF	リンク	H	フォントサイズの変更
FRAME	フレームの挿入	IFRAME	インフレームの挿入

まず付加的な情報を表現するタグについて述べる。META タグには、Web ページ上に直接表現されないページの説明や特徴を現す単語が記述されている場合があり、Web ページの特徴を抽出する際に必要であると考えた。また、検索結果として得られた URL が参照する Web ページが、フレームやインフレームを使用している場合、十分な情報を得ることができない場合が多いことが分かった。このような場合には、フレームやインフレームを参照している URL を得て、同じく分析を行うことで解決した。

次に直接 Web ページ上に表現される文章を修飾するタグについて述べる。HTML ファイルにおいて TITLE タグで修飾されている部分は、Web ページ上ではタイトルを意味し、ページの特徴を表している部分といえる。また、H タグや STRONG タグなどで修飾された部分は、Web ページの作者が強調して表現したかった部分であると考えられるので、ページの特徴を表している部分といえる。

このようにタグを利用して Web ページにおいて作者が強調したい重要な部分と、その他の部分を適切に判断し、それぞれに出現する単語の重要度を変化させることで、Web ページの特徴をより明確にすることができると考えた。

クラスタリング

HTML ファイルを解析した結果を用いてクラスタリングを行う。結果を形態素解析し、Web 文書ごとの単語の出現頻度、文書全体での単語の出現頻度などを、前節で述べた手法を用いて算出する。形態素解析には、奈良先端科学技術大学大学院で開発されたシステム「茶筅」を用いた [21]。次にそれらの情報を基にして、各 Web 文書をベクトル空間法で表現する。これには tf/idf 法を用いた。Web 文書 D_i 中の単語 t_j の出現頻度を f_{ij} とすると、

$$tf_{ij} = \log(1 + f_{ij}) \tag{3.3}$$

となる。対数を用いたのは、$tf_{ij} = f_{ij}$ とすると出現頻度の高い単語に過大な重みを与えてしまうためである。また、1 を足すのは $f_{ij} = 0$ のとき $tf_{ij} = 0$ とするためである。

また d_j を単語 t_j を含む Web 文書の総数、n を Web 文書の総数としたとき、idf_j は、

$$idf_j = \log(\frac{n}{d_j}) \tag{3.4}$$

となる。d_j が小さく、単語 t_j を含む Web 文書が少ないほど値が大きくなることを示す。対数をとるのは、idf_j の値が過度に変化するのを防ぐ目的がある。

単に tf/idf を用いると、長い Web 文書に現れる単語ほど重みが高くなってしまうという問題点がある。そのため Web 文書長の正規化を行った。正規化にはコサイン正規化を用いた。tf_{ij}、idf_j を用いてコサイン正規化による文書長の正規化係数は、以下のようになる。コサイン正規化は、Web 文書全体に含まれる単語の総数を m としたとき、m 次元ベクトル $(tf_{i1}idf_1, tf_{i2}idf_2, \cdots, tf_{im}idf_m)$ の向きを変化させずに、ベクトル長を 1 にする処理であるといえる。

$$n_i = \sqrt{\sum_{j=1}^{m}(tf_{ij} \times idf_j)^2} \tag{3.5}$$

まとめると、文書 D_i 中の単語 t_j に対する重み w_{ij} は、次式で求めることが可能となる。

$$w_{ij} = \frac{tf_{ij} \times idf_j}{n_i} \tag{3.6}$$

こうした計算を行って、Web 文書それぞれに対して m 次元の特徴を表すベクトルが与えられた。これらのベクトルを使って前章で述べた階層的クラスタリングのアルゴリズムを用いて

クラスタリングを行った。各 Web 文書について内積を計算し、最も類似している 2 つの Web 文書を合成して新たなクラスタ (Web 文書) とする処理を繰り返すと、最終的にクラスタリングが終了し、Web ページをノードとする樹形図が完成する。

Web 文書クラスタリング結果の考察

ここでは Web 検索結果をクラスタリングした結果の考察を行う。本システムを用いて「小林」という検索クエリで 50 件の Web ページをクラスタリングした結果を図 3.4に示す。番号はそれぞれ Web ページを表しており、GoogleAPI から取得した順番である。
これを観察すると以下のような問題があることが分かる。

1. ルートからノードの Web ページにはたどり着くまでに階層が深い
2. クラスタにまとまっていない Web ページが存在する

本研究のクラスタリング手法は、すべての Web 文書が強制的にクラスタリングされてしまう。そのために、あまり類似していない Web ページ同士がクラスタリングされ、樹形図が階段状になってしまう場合がある。このような樹形図は階層が深く、そのままユーザに提示しても煩雑で分かりにくくなってしまう。
このクラスタリングにおける問題を概観提示部において表示インタフェース的に解決するために、うまくまとまっていない階段状になっている部分をまとめて新たなクラスタとすることを考えた。このようにすることで深い階層を浅くすることが可能となり、概観提示部においてユーザに対して検索結果の概観をより分かりやすく提示することができる。

3.4.2 概観提示部

概観提示インタフェースは、ユーザの入力した検索クエリに対しての検索結果を検索結果のクラスタリング部でクラスタリングし、概観提示部でクラスタリング結果を Hyperbolic Tree を用いてユーザに提示する。以下に概観提示部の詳細について述べる。

概観提示画面

図 3.5に本システムにおいて「小林」という検索クエリで検索を行った場合の概観提示画面を示す。類似ページから構成された部分木は、中央に近いものほど大きく表示され、中央から遠いものほど小さく表示されていることが分かる。ちょうど半球の表面に樹形図を貼り付け、上から眺めたようなイメージである。ユーザは必要と思われる部分木をドラッグして中央に移動させることにより部分木を拡大して見ることができる。子ノードを持たないノードは Web ページのタイトルを表し、子ノードを持つノードは子ノードに対するラベルを表現している。
次に各ノードの関係について述べる。単語 1〜6 と Web ページ A,B,C が図 3.6のように接続

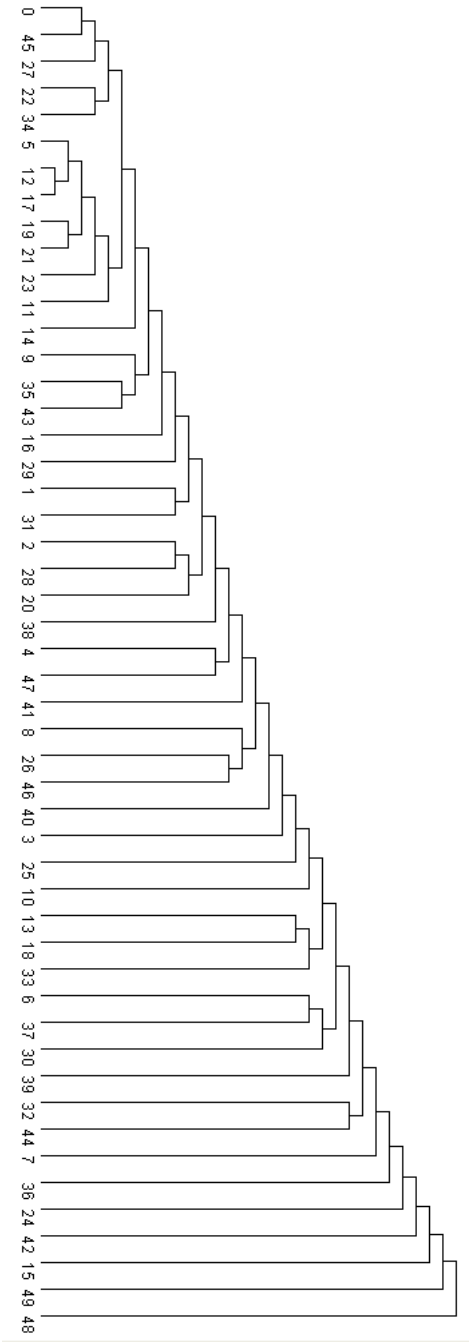

図 3.4: 「小林」という検索クエリで 50 件の Web ページをクラスタリングした結果

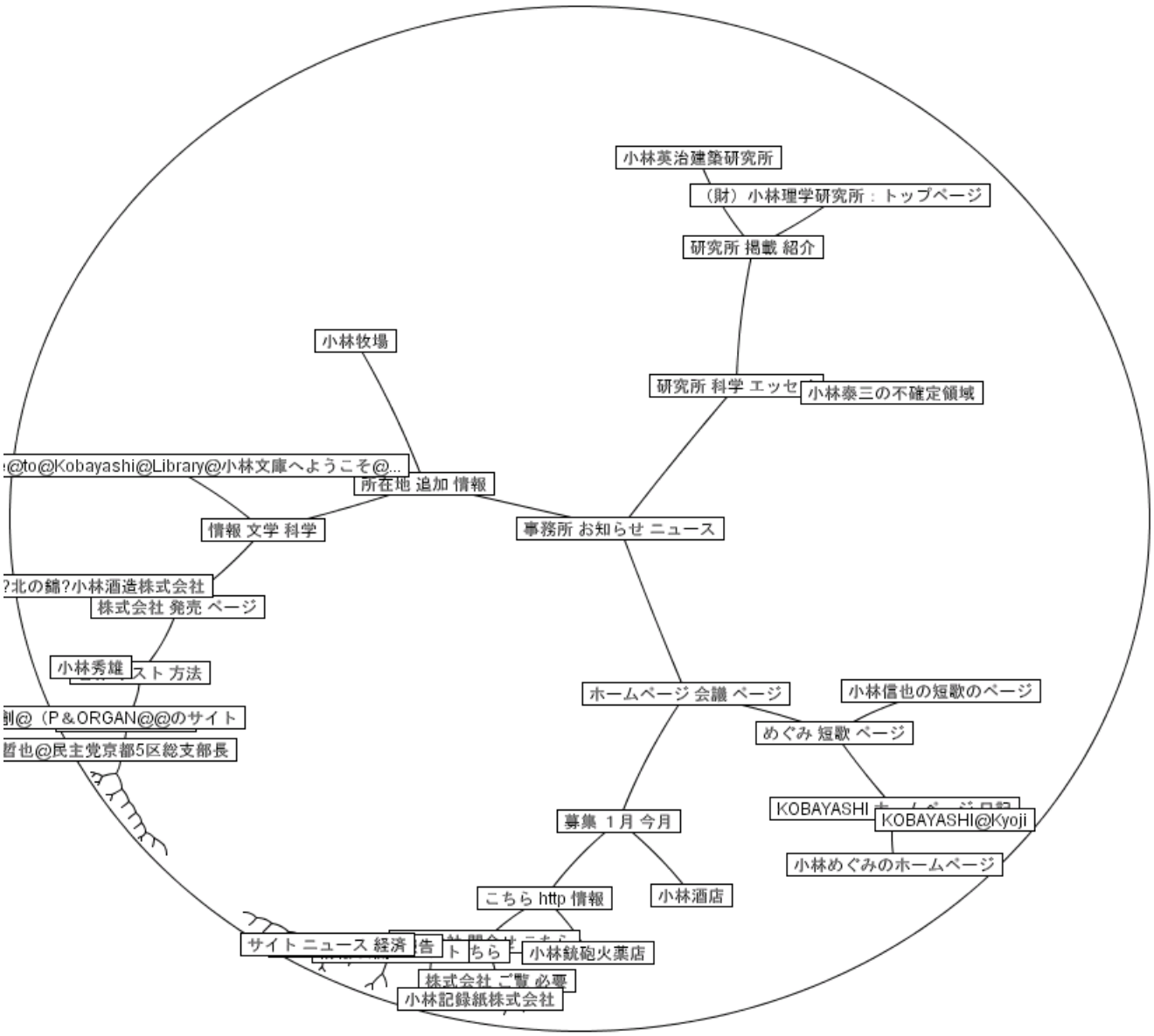

図 3.5: システムの提示画面

されている場合を考える。

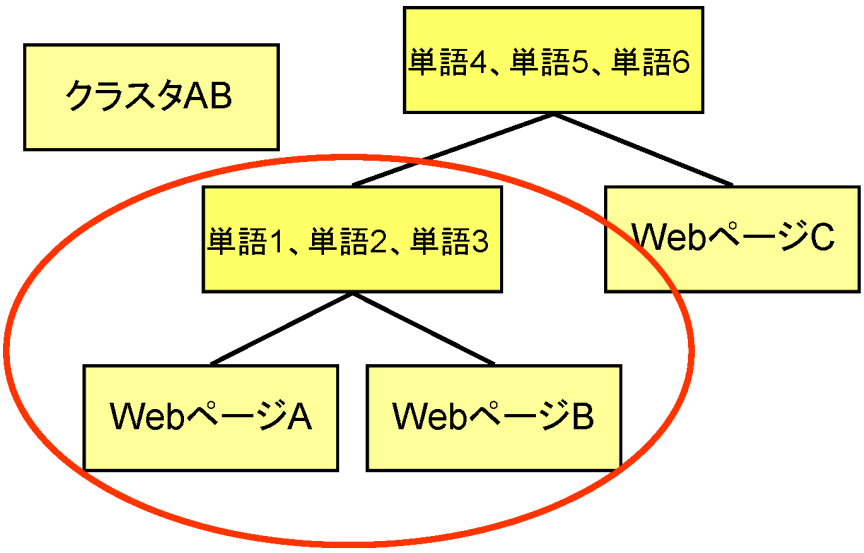

図 3.6: Web ページ A と Web ページ B の関係

クラスタリング部において類似度を求める際に、Web ページ A と Web ページ B の関係で特に結びつきが強い上位 3 単語を Web ページ A と Web ページ B の親ノードとした。これらの単語を Web ページ A と Web ページ B から構成されるクラスタのラベルとしてユーザに提示する。図 3.6において Web ページ A と Web ページ B は親ノードである単語 1, 単語 2, 単語 3 に対して強い関連があることを示している。さらに Web ページ A と Web ページ B からなるクラスタ AB(図 3.6中の赤丸で囲まれた部分) と Web ページ C は、単語 4, 単語 5, 単語 6 に対して強い関連があることを示している。ユーザはこれらのラベルを見ながら必要な情報が記述されている Web ページを選択することができる。

図 5.1にクラスタリングが有効に行われている部分を示す。
「小林」というクエリによって得られた検索結果をクラスタリングし、議員のホームページがそれぞれの近傍に配置されていることが分かる。

クラスタリングにおける問題点の改善

本研究のクラスタリング手法には、あまり類似していない Web ページ同士がクラスタを形成してしまい樹形図が階段状になってしまう場合があるという問題点がある。この問題を表示インタフェース的に改善するため、それらの Web ページを「その他」というラベルをつけた 1 つのクラスタにまとめることで樹形図を変形してユーザにとってより見やすい提示画面を提供した。

図 3.8の左図はあまり類似していないにもかかわらずクラスタリングが行われている部分を表している。この部分は樹形図的に階段状になってしまっていて、そのまま表示してしまう

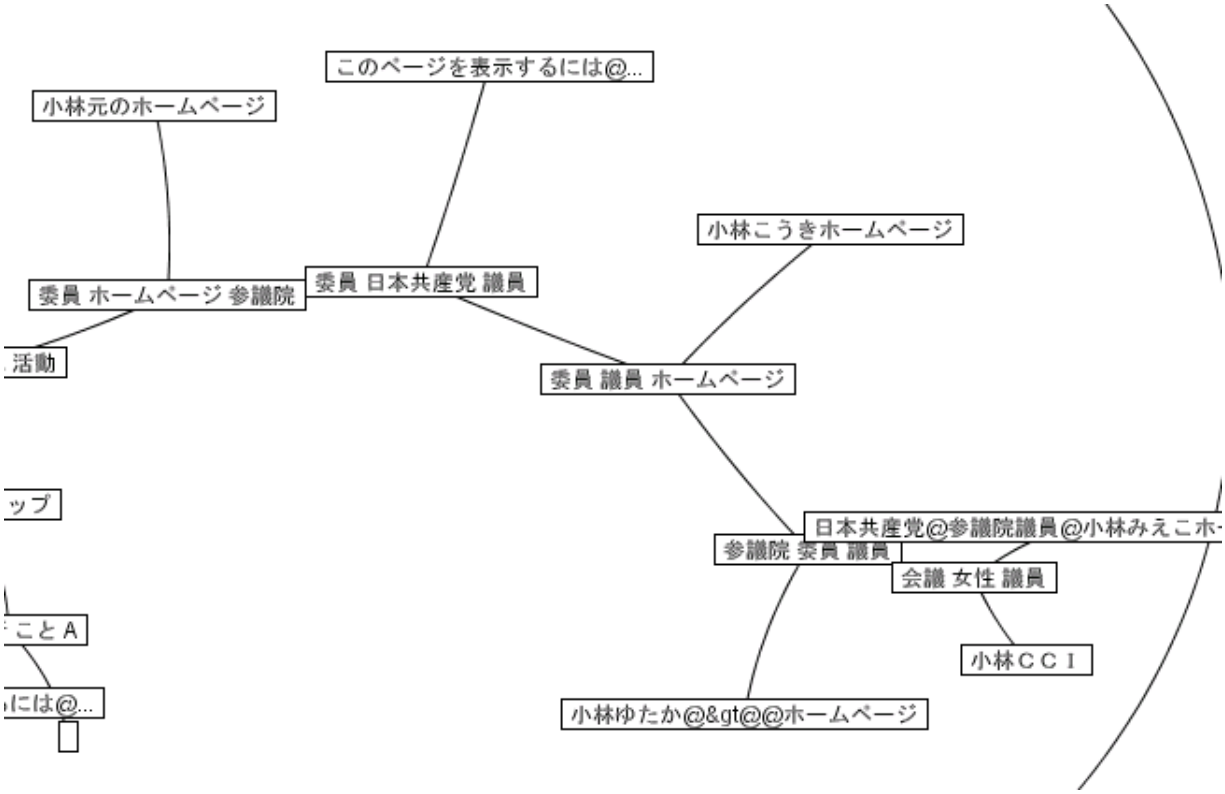

図 3.7: クラスタリングが有効に行われている部分木

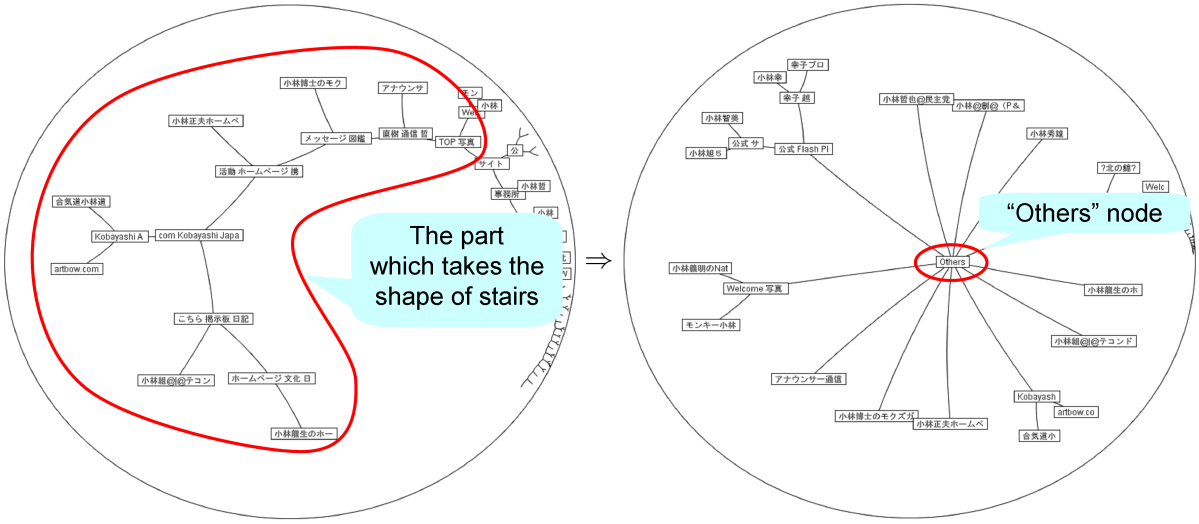

図 3.8: 表示インタフェース的なクラスタリングの改善

とユーザの混乱を招く恐れがある。右図は階段状になり適切にまとまっていない Web ページをまとめて「その他」ノードの子ノードにした場合を表している。右図は左図に比べて無駄なラベルが省かれており、代わりに表示できる Web ページの数が多くなっていることが分かる。このような表示インタフェースによってユーザはより Web 検索結果全体の概観がつかみやすくなり、効率よく情報を収集することが可能となる。

3.4.3 ページ重要度の表現

本研究では検索結果の取得を GoogleAPI を用いて行っている。Google は Web ページの重要度を計算する方法としてページランクを採用している。ページランクを用いることで「多くの良質なページからリンクされているページは、やはり良質なページである」という再帰的な関係を基に相対的な Web ページの重要度を求めることが可能である [22]。

概観提示インタフェースは検索結果をクラスタリングした結果を Hyperbolic Tree を用いて提示する。このため、Google の提示インタフェースのようにページの重要度による順序付けができない。そこで、本システムはノードの色の濃淡を利用してユーザにページの重要度を提示する。具体的には重要度が高いものほどノードの色は濃く、重要度が低いものほどノードの色は薄く設定される。このようにすることで重要度による順序付けができないという問題点を改善し、ユーザはページの重要度を視覚的に理解することが可能となるり、膨大な検索結果の中から重要度の高い Web ページを見つけることが容易になる。さらに、ユーザは本インタフェースの提供する色の濃淡情報を用いることで検索結果の全体像をより直感的に理解することが可能となるのである。図 3.9に色付けを行った概観提示インタフェースを示す。

ここで本章のまとめとして、概観提示インタフェースのシステムの流れと相互関係を図 3.10にまとめておく。

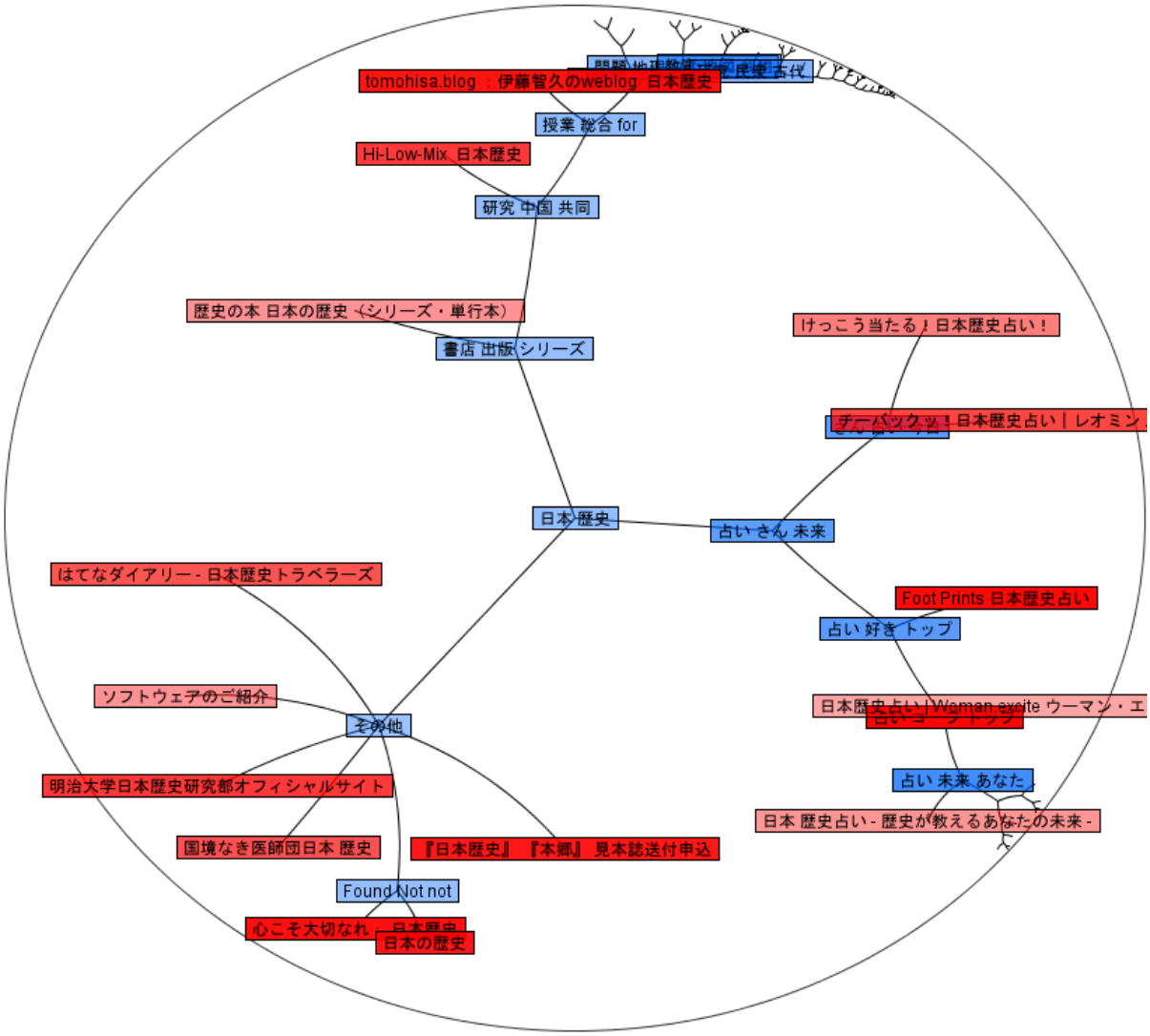

図 3.9: 色づけを行った概観提示インタフェース

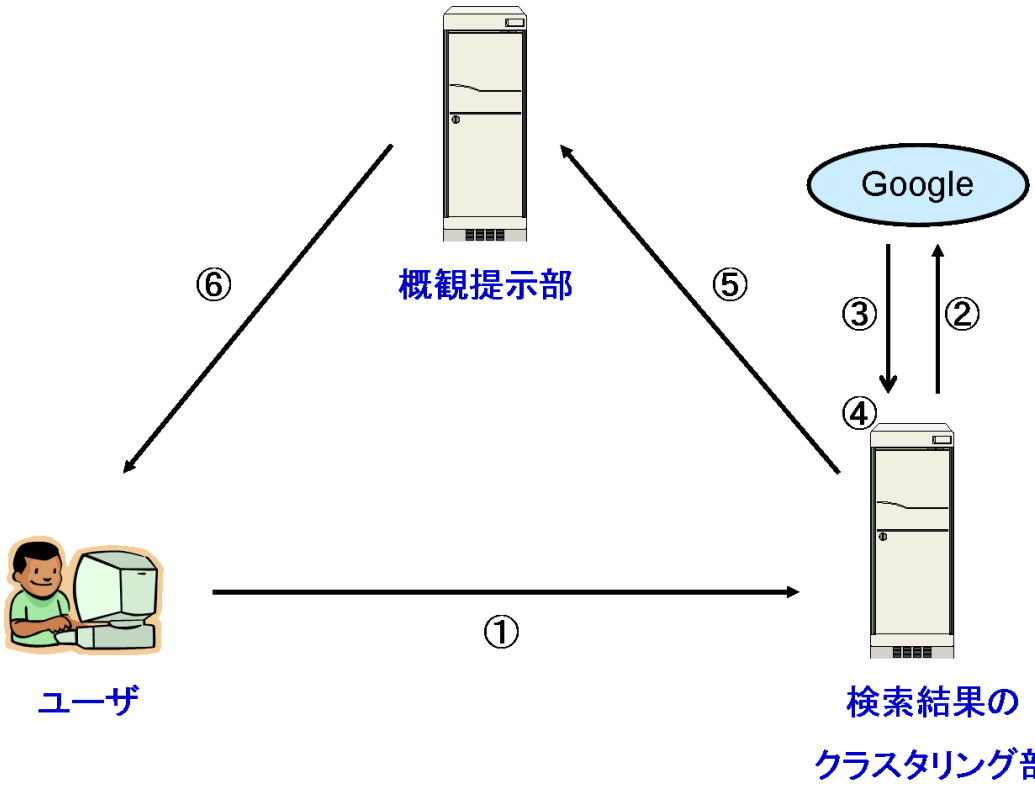

図 3.10:① 検索クエリ,②GoogleAPI,③ 検索結果,④Web ページの分析とクラスタリング,⑤ クラスタリング結果,⑥ 概観提示

第4章　概観提示インタフェースの機能拡張

本章では情報指向型検索を行う際の既存インタフェースの情報の収集と保存・情報の利用を行う際の問題点を解決するための概観提示インタフェースの機能拡張について述べる。

4.1　HTML のタグ情報を利用した Web ページの要旨の抽出

既存検索インタフェースを用いて情報指向型検索を行い情報を収集する場合、ユーザは提示された検索結果のタイトルと Web ページの抜粋、または実際の Web ページを閲覧して必要な情報かどうかを判断する。この際に、実際の Web ページを見て判断を行う場合、Web ページを閲覧しては検索結果提示画面に戻る、という作業を何度も繰り返す必要があり、情報指向型の検索を効率よく行えるとはいえない。

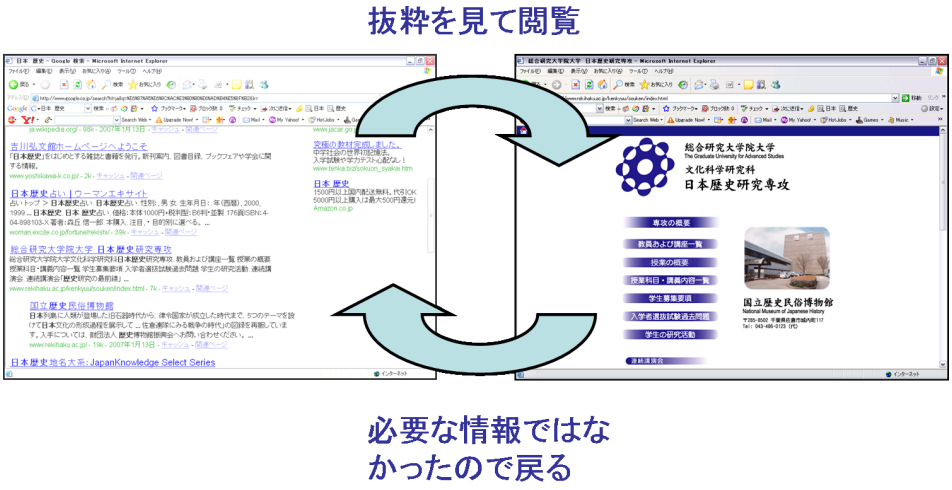

図 4.1: 検索結果提示画面と Web ページの往復

このため、実際の Web ページを閲覧し、画面を切り替える必要性が少なくなるように、Web ページのタイトル以外の情報を付加してユーザに提示することが重要となる。しかしながら、既存インタフェースが与えている情報はキーワードを含む Web ページの抜粋などであり読みにくく Web ページの内容を適切に表現していないために Web ページの内容をうまく理解することができない場合があるなどの問題がある。図 4.2は既存検索インタフェースが提示する Web ページの内容の例である。

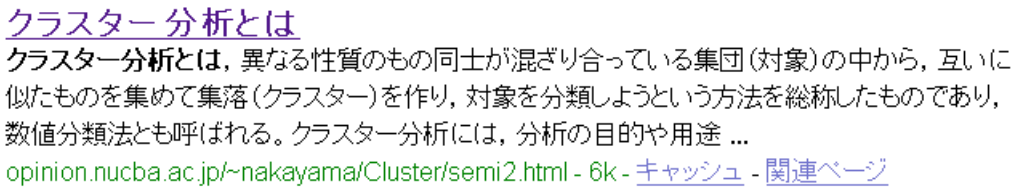

図 4.2: 既存検索インタフェースの提示する Web ページの情報例

本研究では、このような問題を解決するために HTML のタグ情報を利用して Web ページの要旨を抽出し、ユーザにわかりやすい適切な方法で提示する。例えば H タグが修飾している部分は見出しであり、Web ページの内容を良く表した文が記述されると考えられる。図 4.3は H タグを利用して図 4.2の Web ページから要旨を抽出した場合の例である。図 4.2と図 4.3を比較した場合、タグから Web ページの要旨を抽出した図 4.3の方が Web ページの内容をよりよく表現していると言える。

このような Web ページの内容を顕著に表現したいくつかの文を要旨としてユーザに提供する。要旨から Web ページの内容を適切に理解することができれば、ユーザは実際の Web ページを閲覧することなく必要な情報を取捨選択することが可能となる。

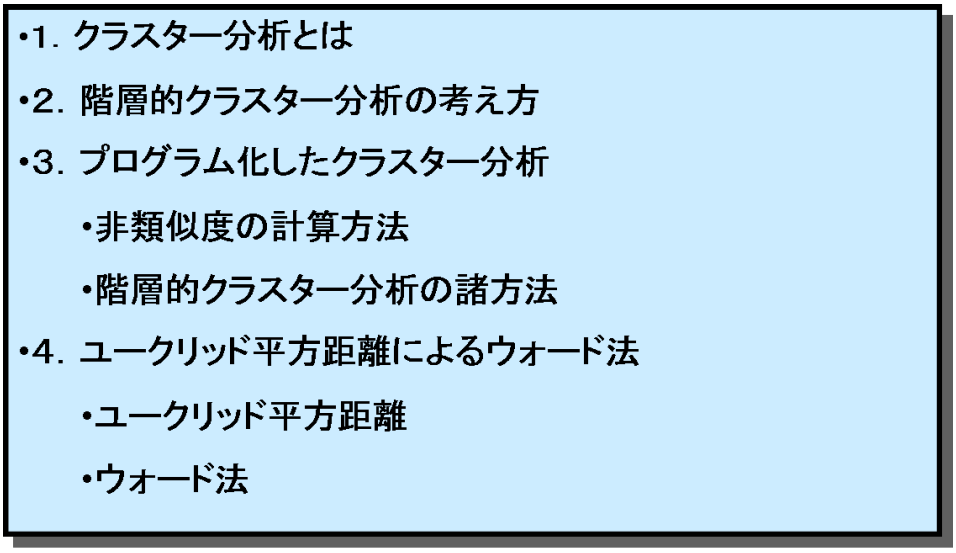

図 4.3: Hタグを用いた見出し抽出の例

4.2　要旨の提示法

図 4.4はタグを利用して得た要旨を提示した例である。ユーザは要旨を閲覧したい Web ページのノードを中央に移動させ、クリックすることで検索結果提示画面からの切り替えを行わずに Web ページの要旨を閲覧することができる。要旨を見て Web ページ中に必要な情報が存在すると判断した場合には実際の Web ページを閲覧し、情報の収集を行う。

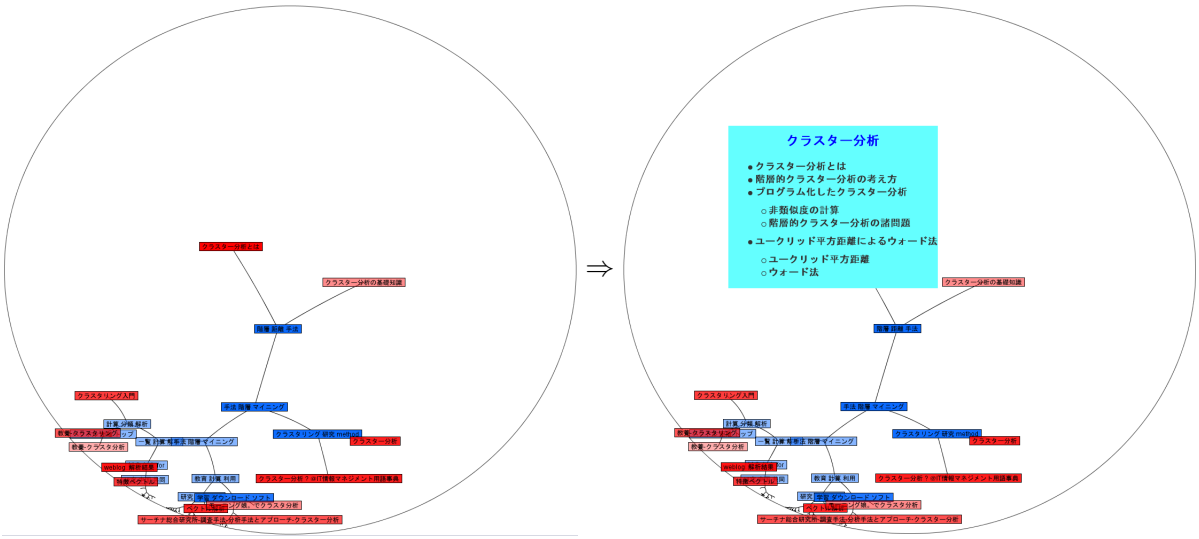

図 4.4: 要旨の提示

4.3　情報の保存

複数の Web ページを巡回して情報を収集していく情報指向型検索を行う場合、図 4.6のように、Web ページの必要な箇所を保存しておきスクラップブックのようなものを作成し、後にそれを利用できれば便利である。既存検索インタフェースを用いてこのような作業を行う場合には、Web ページ全体を保存してフォルダに整理したり、必要な箇所をメモするなどの作業が必要となる。しかし、このような作業は情報検索による情報の収集と情報の保存が一連の作業として行われないため、情報指向型検索をスムーズに行うことが難しいと考えられる。

そこで本研究ではこのような問題点を解決するために、概観提示インタフェースにおける検索機能と情報保存機能を組み合わせることで、情報の収集と情報の保存を一連の作業として行うための機能拡張を行った。ユーザは Web ページを閲覧し、図 4.7のようにして必要な任意の箇所をマウスドラッグにより指定し、スクラップとして取り出すことができる。図 4.8はシステムを用いて Web ページのスクラップを取り出した例である。また、本研究では Web ページのスクラップを画像として抽出している。そこで、スクラップの保存時にユーザが重要であると考えた部分を編集して保存が行えるように実装した。図 4.9は図 4.8のスクラップを編集して保存したものである。 このようにして保存したスクラップから図 4.10のようなスクラッ

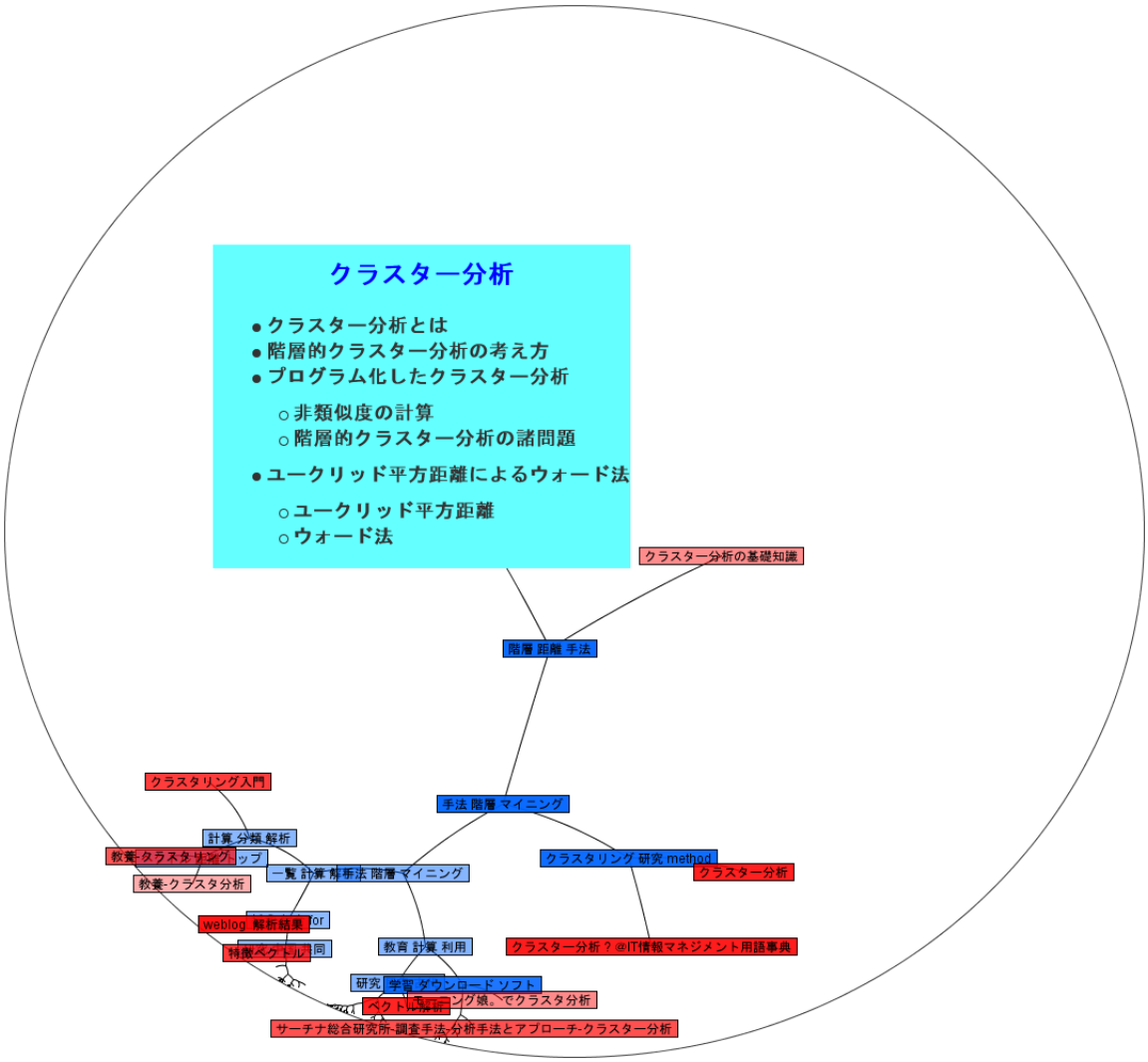
クラスター分析
クラスター分析とは
階層的クラスター分析の考え方
プログラム化したクラスター分析
非類似度の計算
階層的クラスター分析の諸問題
ユークリッド平方距離によるウォード法
ユークリッド平方距離
ウォード法
クラスター分析の基礎知識
階層 距離 手法
クラスタリング入門
手法 階層 マイニング
計算 分類 解析
クラスタリング 研究 method
クラスター分析
教養-クラスタ分析
weblog 解析結果
教育 計算 利用
クラスター分析？@IT情報マネジメント用語事典
研究
学習 ダウンロード ソフト
サーチナ総合研究所-調査手法-分析手法とアプローチ-クラスター分析

図 4.5: 要旨の提示

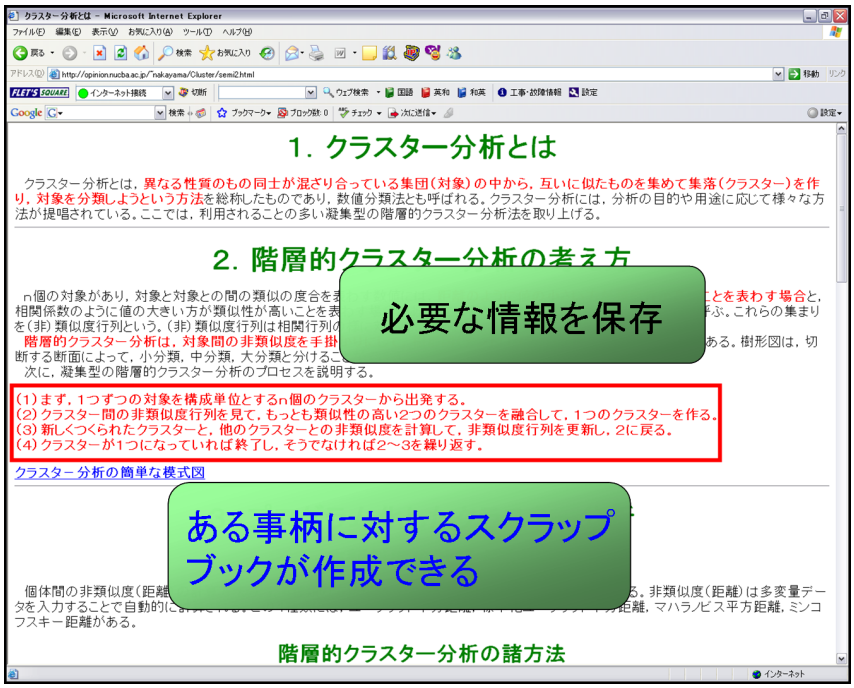

図 4.6: Web ページの一部を保存

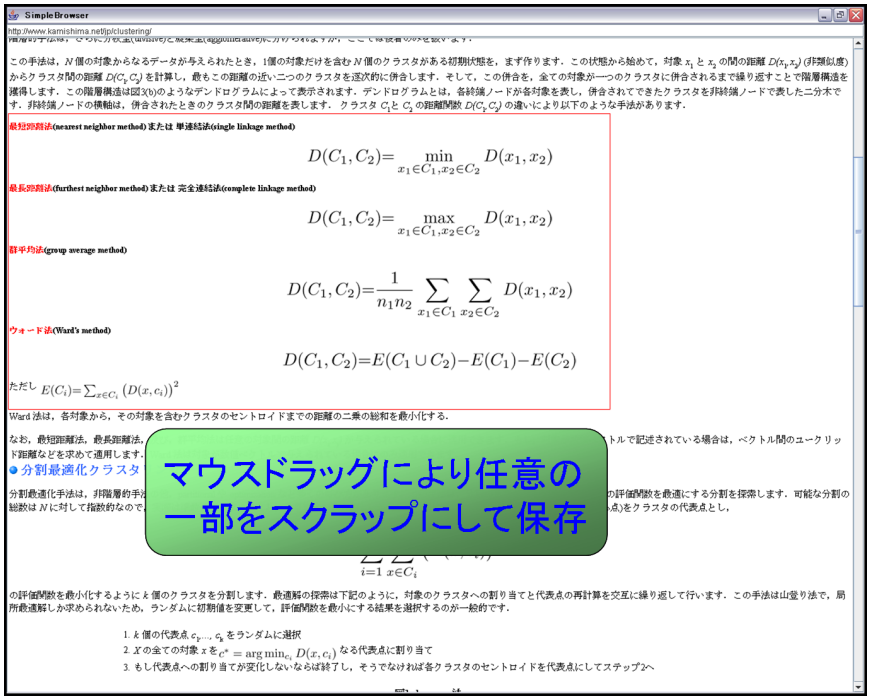

図 4.7: スクラップの保存

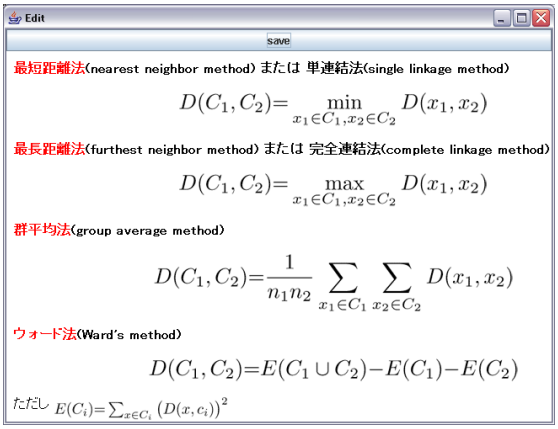

図 4.8: スクラップの例

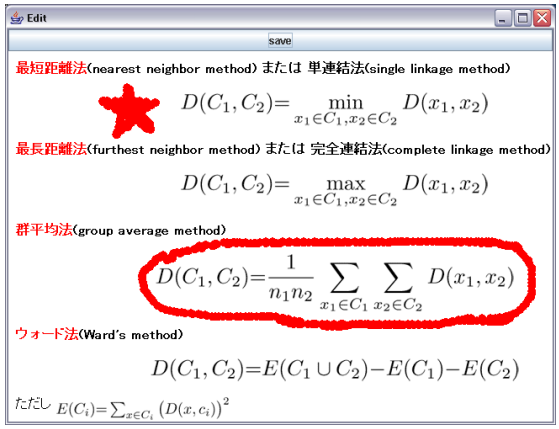

図 4.9: 編集したスクラップの例

プブックを保存することができる。スクラップブック中のスクラップはユーザが自由に位置を変えることが可能であり図 4.10のように任意のラベルを用いて収集したスクラップをユーザが後に利用しやすいように整理して保存することができる。

4.4 情報の利用

以上のようにして得られたスクラップブックをレポート作成などの具体的なタスクでユーザは本システムを利用することができる。スクラップブックに保存された画像は図 4.11のように拡大縮小が可能になっており、収集したスクラップの中から必要なスクラップを利用してレポートの作成などを容易に行うことが可能となる。

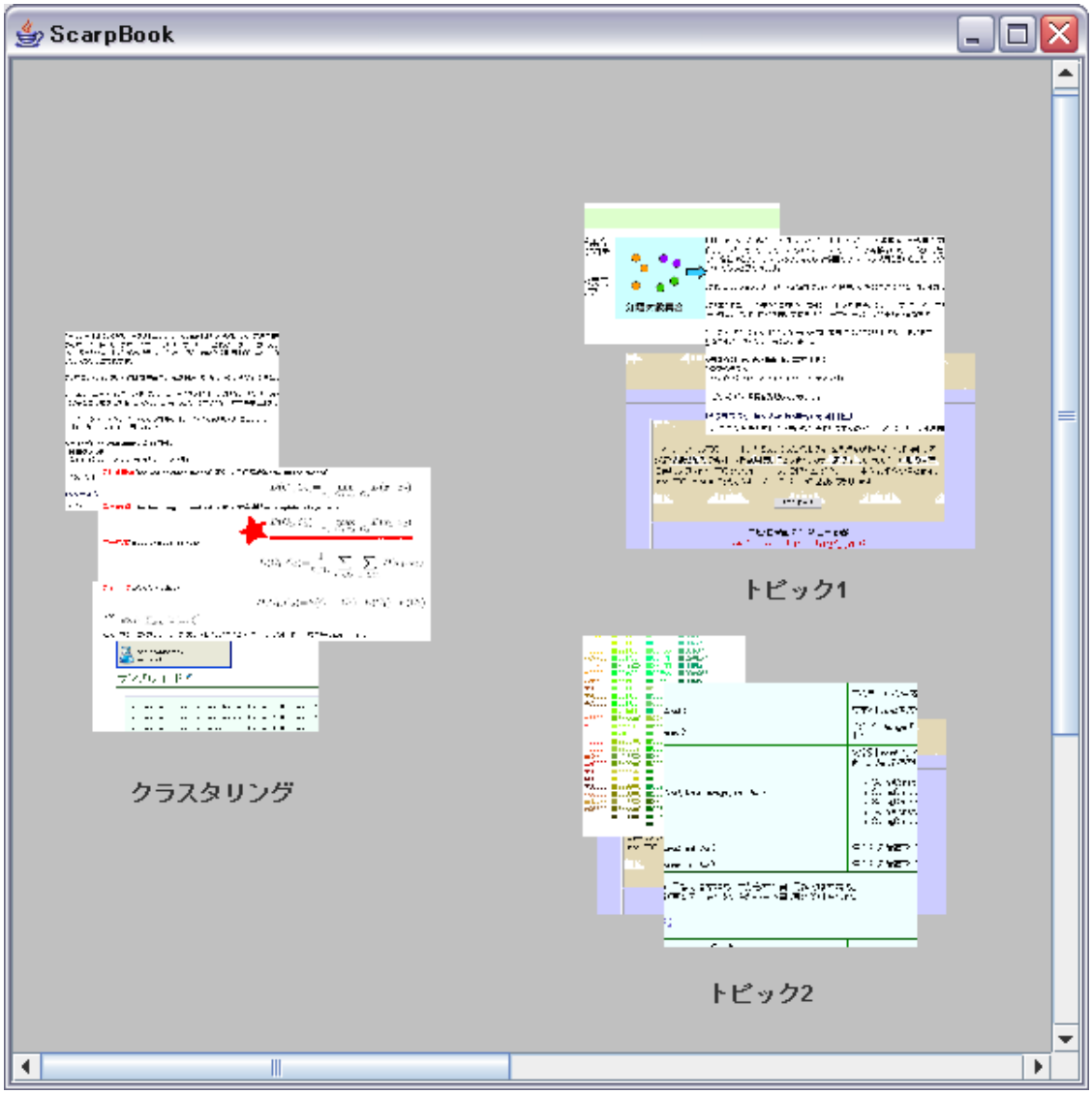

図 4.10: スクラップブック

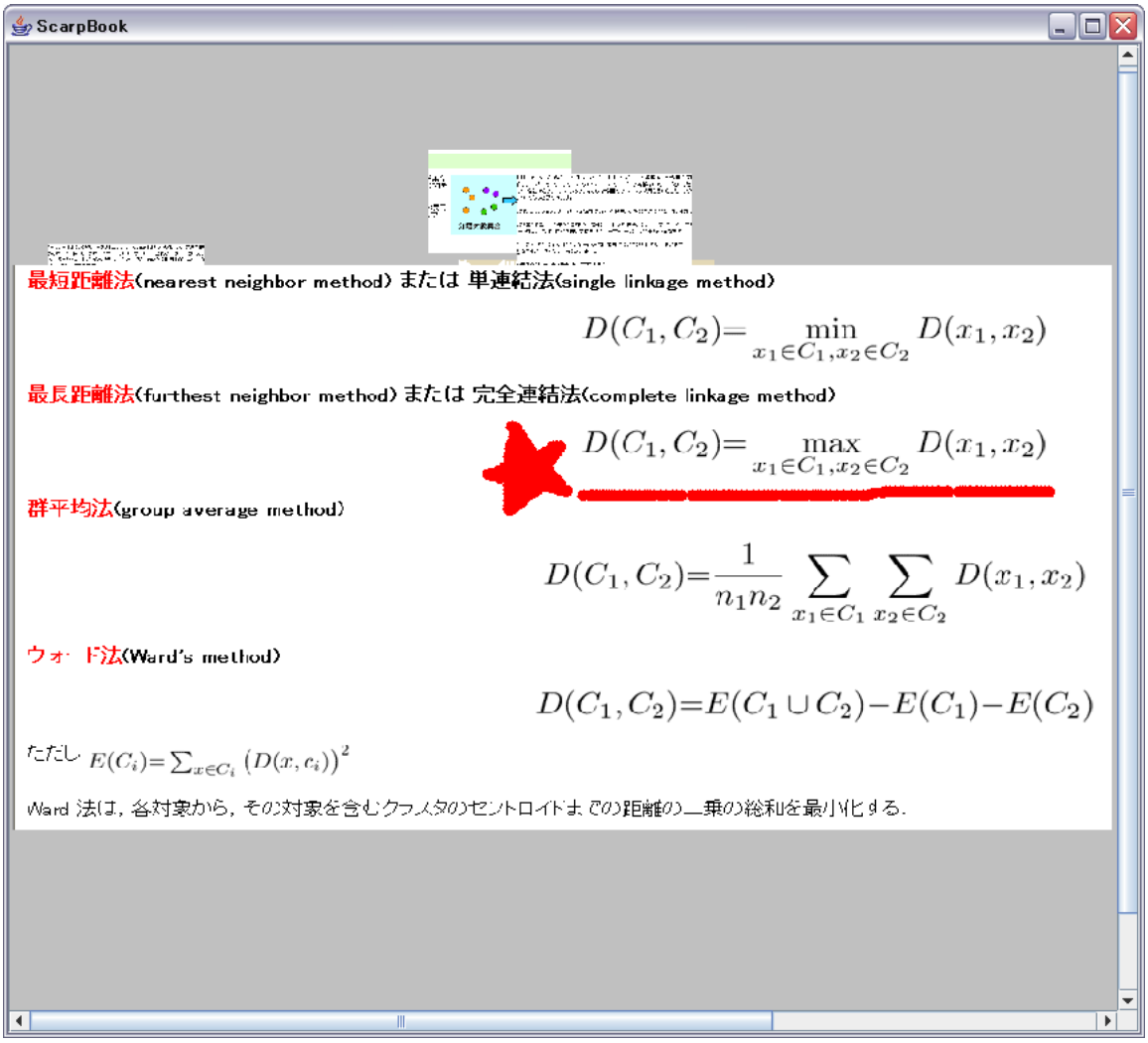

図 4.11: スクラップブックの利用

第5章　インタフェースの実用例

ここでは具体的なタスクの例を挙げて本インタフェースの実用例について説明する。あるユーザが日本の歴史について様々な観点から調べたレポートを作成したいと考えた場合を想定する。この場合、ユーザはインターネットを用いて情報指向型検索を行い、情報を収集・保存し、後に保存した情報を元にレポートを完成させることになる。

まずユーザはシステムに対して検索クエリ「日本　歴史」を与える。システムはアルゴリズムにしたがって検索結果を分析およびクラスタリングし、その結果をユーザに提示する。図5.1はシステムに検索クエリ「日本　歴史」を与えた場合の提示画面である。このように提示された情報の中からユーザはマウスドラッグにより注視点を変更し、与えられたラベルを参考に類似ページがまとまっている部分木をたどりながら必要な情報を取捨選択することが可能である。図5.2からも、クラスタリングによって類似ページが部分木にまとまっていることが分かる。また、ノードの色の濃淡を利用したページの重要度の提示によって、重要度が高いWebページのノードほどノードの色は濃く、重要度が低いWebページのノードほどノードの色は薄く設定されるため、検索結果中の良質なページを直感的に理解することができるようになっている。

図5.1を見ると右下の部分木は「日本」と「歴史」という単語を含む占いについてのページが集合していることが分かる。もし一般の検索エンジンで検索した場合「日本」と「歴史」という単語が含まれるページで、検索エンジンが重要と判断したページから順に表示する。そのためにユーザの意図とは違うページが多数含まれてしまう場合がある。この例では日本の歴史について調べたいというユーザの意図に反して意図していない占いのページが多数検索結果のリスト中に紛れ込んでしまう。一次元のリストを順に見ているユーザにとってこれは目障りであり、情報収集の効率を阻害していると考えられる。本研究の概観提示インタフェースではこのような問題を解決している。ユーザは意図していない占いに関するページが右下の部分木に集合していることが一目で分かるため、余計なページに情報収集を阻害されることはない。

図5.1の左下の部分木を見ると「その他」を親ノードとするWebページが複数ある。ここにはアルゴリズムではクラスタリングしきれなかったWebページが集合している。検索結果のWebページの中では特殊なWebページであるといえる。

図5.1の上の部分木はかなり大きなものとなっている。ユーザはラベルを見ながら部分木を辿り情報を収集することができる。図5.3は「日本　歴史」を含み「地理」に関するWebページが集合している部分木である。図5.4は「日本　歴史」を含み「教科書」に関するWebページが集合している部分木である。さらに詳しく見ると従軍慰安婦と教科書の問題について述

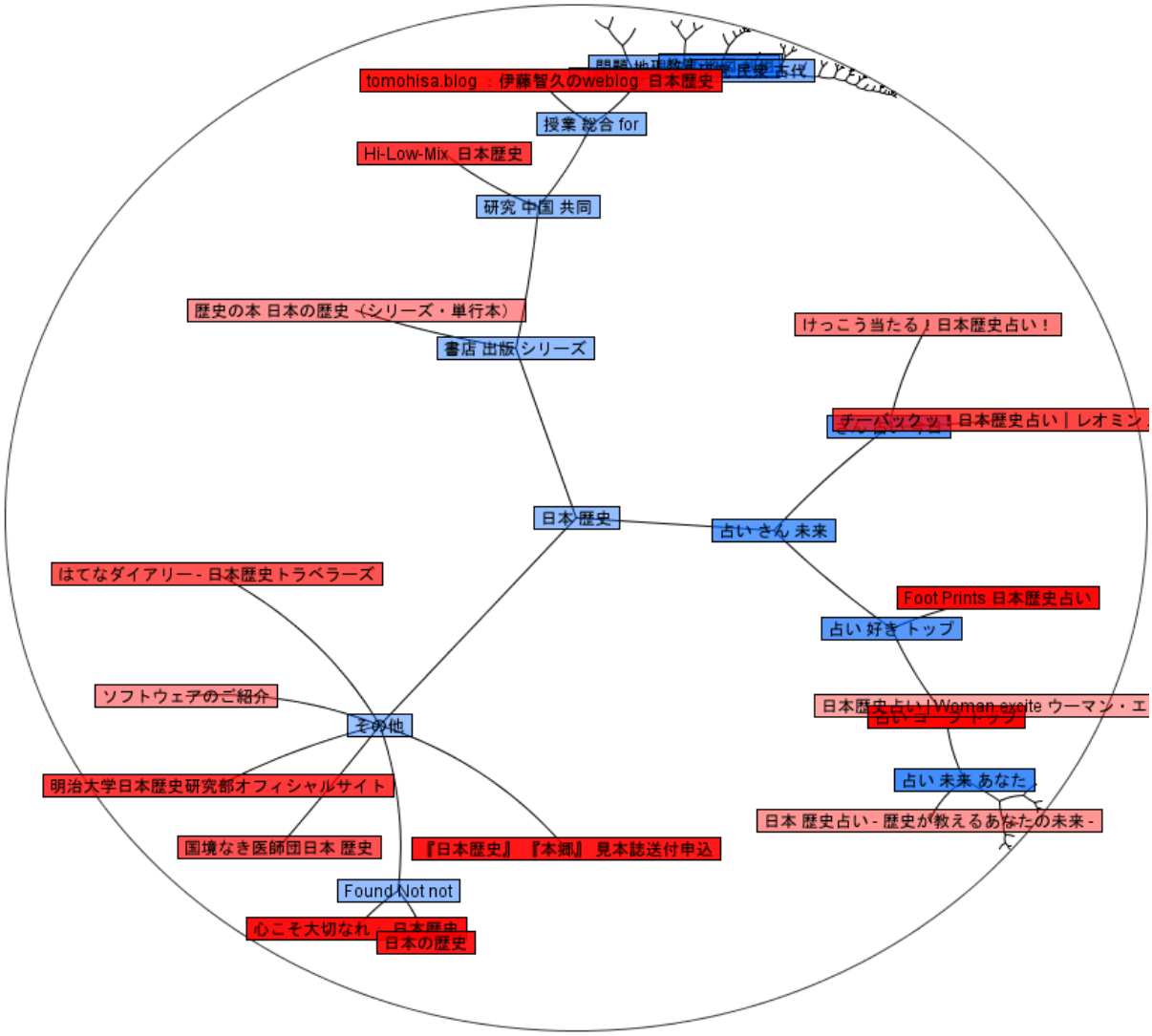

図 5.1: 検索クエリ「日本　歴史」に対するシステムの提示画面

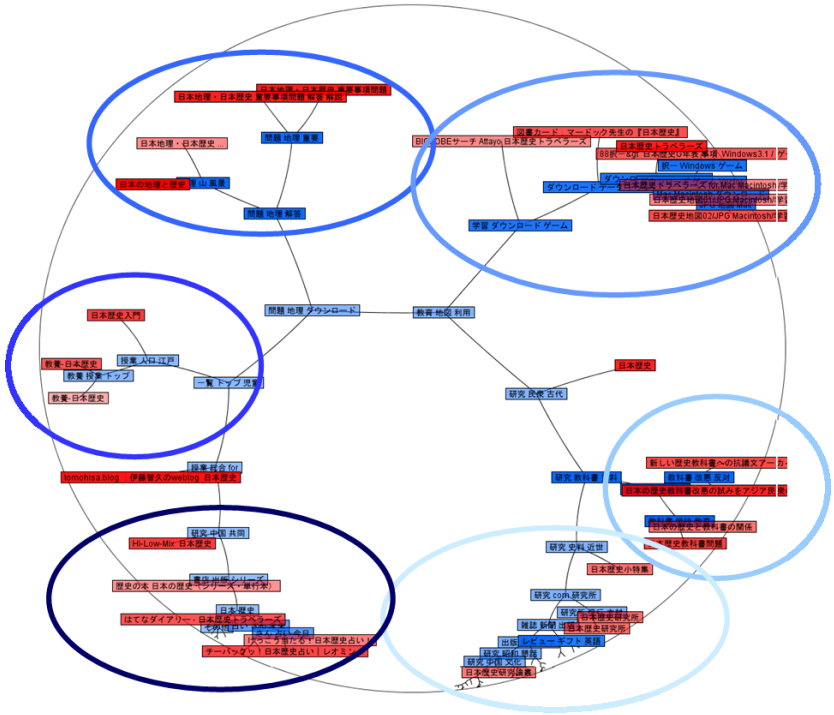

図 5.2: クラスタリングの分布

べているページが多いことが分かる。図 5.5は「日本　歴史」を含み「学会や論文」に関する Web ページが集合している部分木である。このように部分木を辿りながら様々なジャンルの部分木から情報を収集することができる。

ユーザはこのようにして提示される部分木の中からラベルを参考に Web ページを探索し、図 5.6のように、機能追加で行った要旨を閲覧することができる。検索結果提示画面と画面を切り替えることなく Web ページの内容を理解することが可能なため、情報指向型の検索をスムーズに行うことが可能である。

要旨を見て必要な情報を含む Web ページだと判断した場合、ユーザは必要な Web ページの一部分をスクラップとして保存することができる。その際に重要な部分に任意の書き込みを行ったものをスクラップとして保存することも可能である。スクラップブックに保存する際にはスクラップの位置などは任意にユーザが決定することができるため、後にスクラップブックを利用する際に便利なように自由に並べることが可能である。図 5.7はこのようにして作成した「日本の歴史」に関するスクラップブックである。この例では「地理」「教科書」「学会　論文」に関するスクラップを収集し、ラベルをつけて整理している。

保存したスクラップブックを元にしてユーザは後に日本の歴史に関するレポートの作成を効率よく行うことが可能である。図 5.8、図 5.9は作成したスクラップブックの利用例である。ユーザは閲覧したいスクラップを拡大して見ることができる。

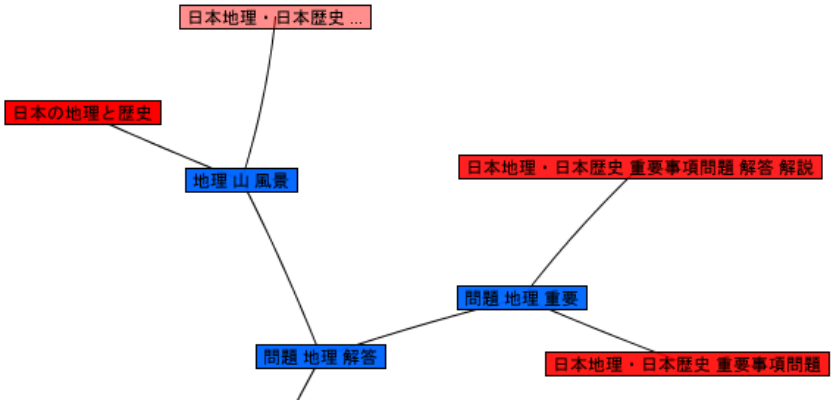

図 5.3: 地理に関する部分木

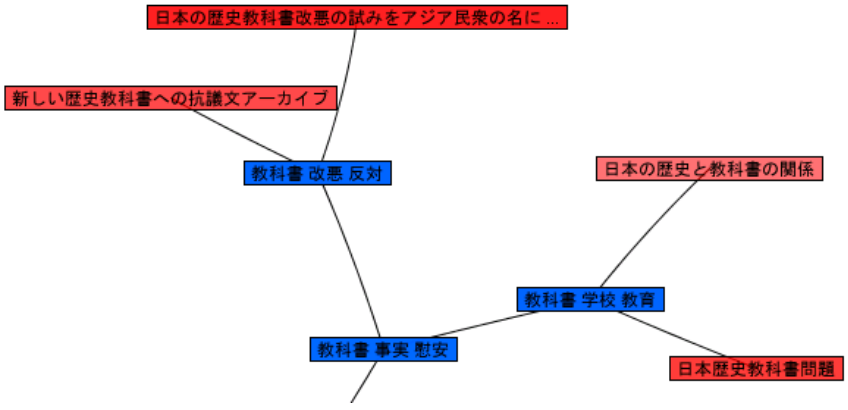

図 5.4: 教科書に関する部分木

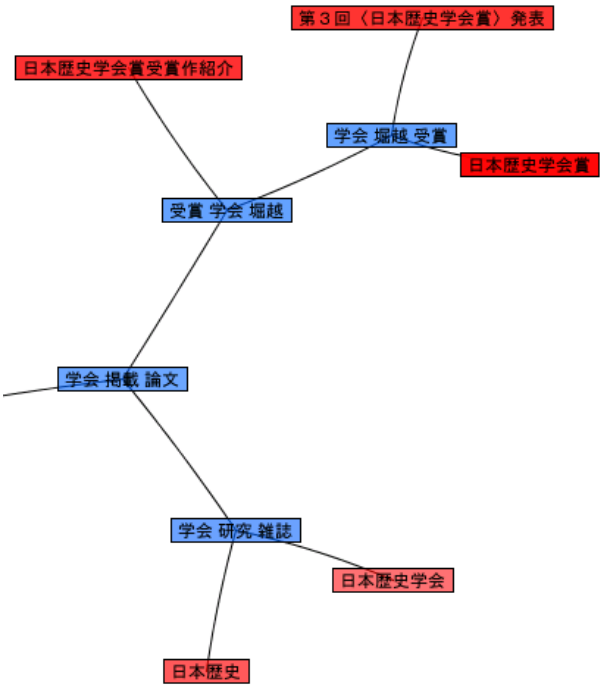

図 5.5: 学会や論文に関する部分木

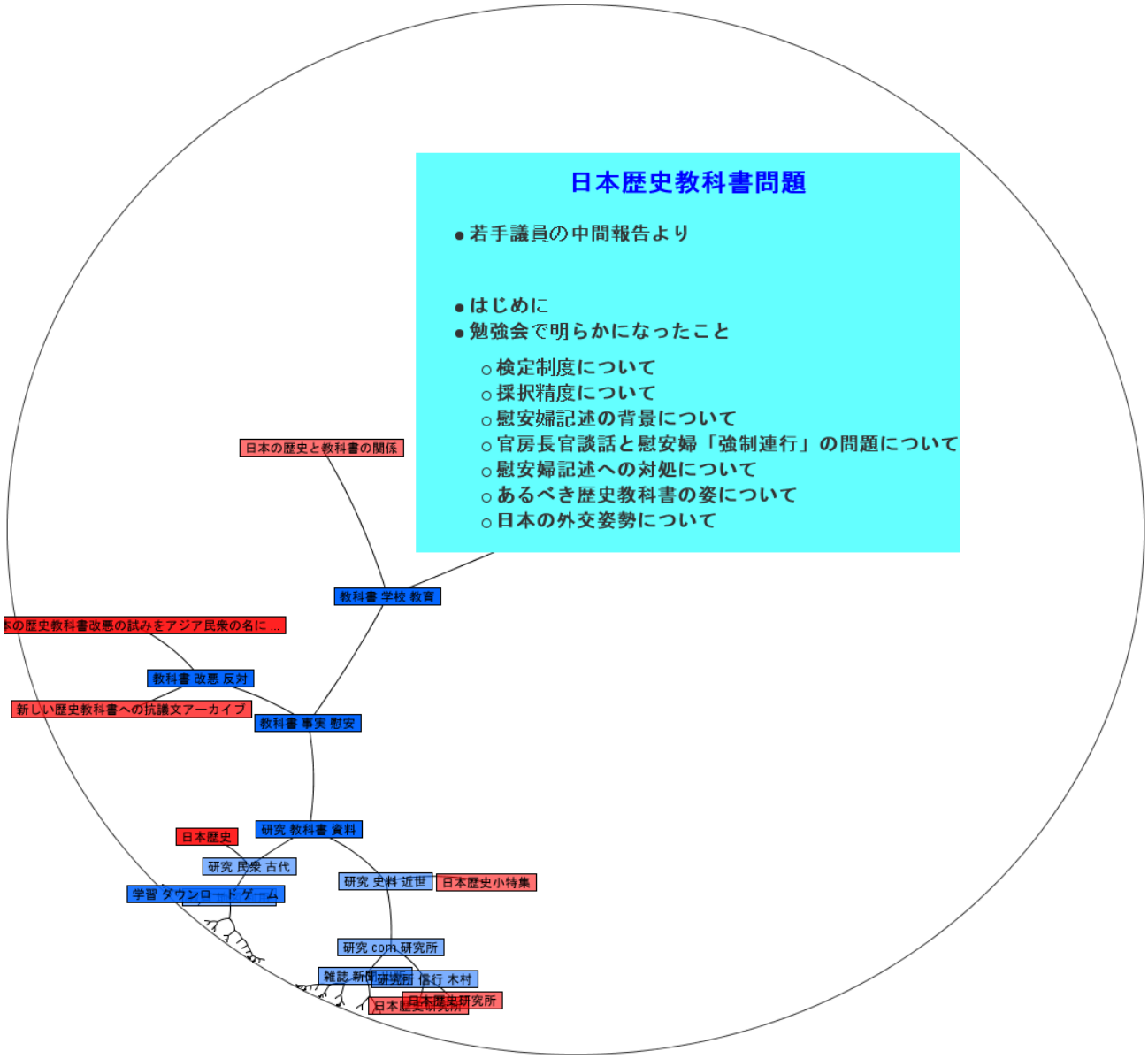
日本歴史教科書問題
若手議員の中間報告より
はじめに
勉強会で明らかになったこと
検定制度について
採択精度について
慰安婦記述の背景について
官房長官談話と慰安婦「強制連行」の問題について
慰安婦記述への対処について
あるべき歴史教科書の姿について
日本の外交姿勢について
日本の歴史と教科書の関係
教科書 学校 教育
本の歴史教科書改悪の試みをアジア民衆の名に ...
教科書 改悪 反対
新しい歴史教科書への抗議文アーカイブ
教科書 事実 慰安
研究 教科書 資料
日本歴史
研究 民衆 古代
研究 史料 近世
日本歴史小特集
学習 ダウンロード ゲーム
研究 com 研究所
雑誌 新聞
研究所 信行 木村
日本歴史研究所

図 5.6: 要旨の提示

図 5.7: スクラップを保存

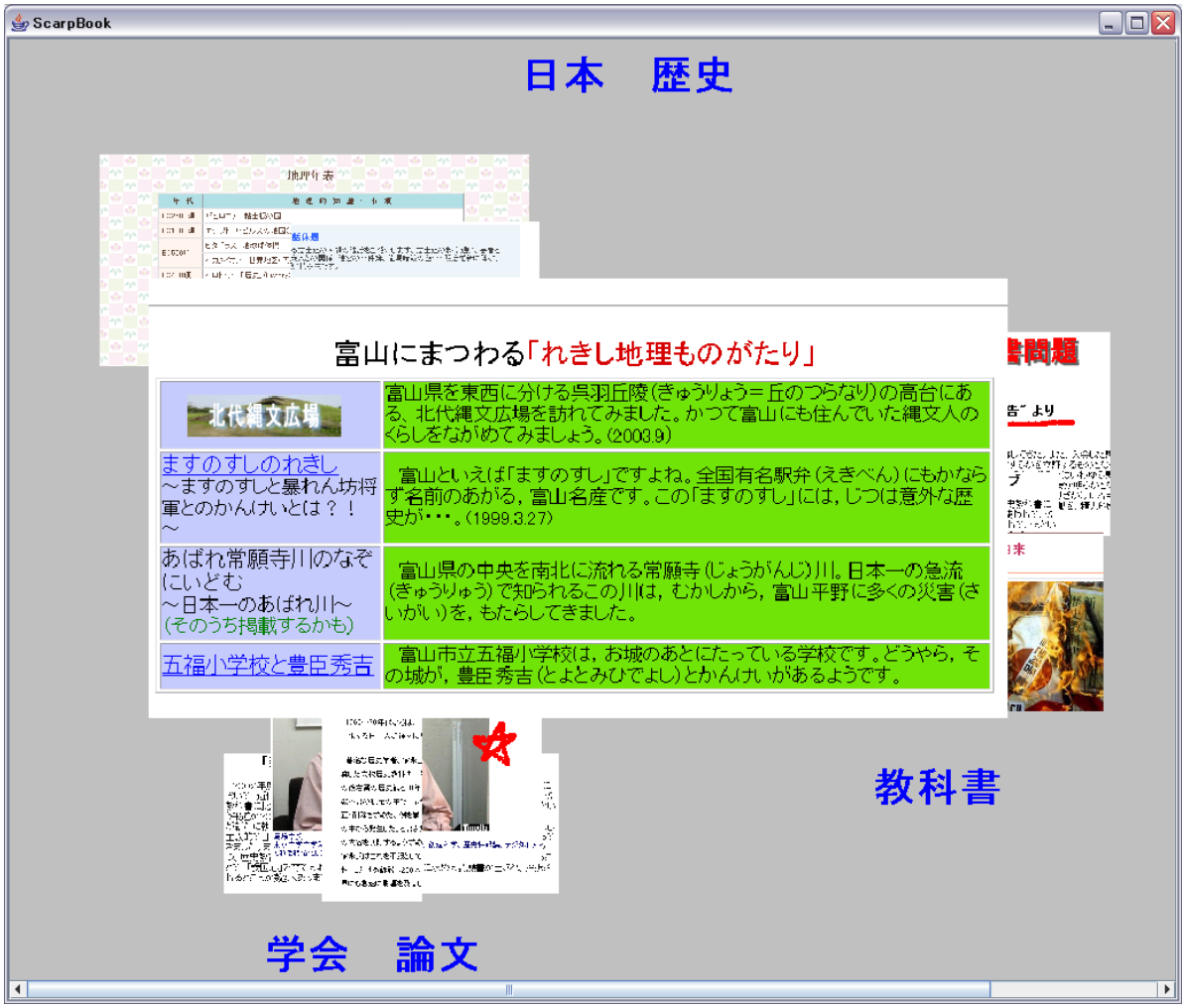

図 5.8: スクラップブックの利用 1

図 5.9: スクラップブックの利用 2

第6章　考察

6.1　インタフェースの考察

6.1.1　Hyperbolic Tree を用いた提示法について

一次元のテキストリストを用いて検索結果を提示するインタフェースではユーザは様々なジャンルの Web ページを含むリストに対して、1 つの Web ページを単位に順に確認する必要があるのに対して、本インタフェースでは類似した Web ページによって構成される部分木を単位に見ていけばよい。そのため様々なジャンルから情報を収集しなければならない場合にユーザは思考を切り替えることなく部分木ごとに必要な情報を効率よく取捨選択することができるのである。また、検索結果が 1 画面に納められていることで、ユーザは部分木を参考にして容易に検索結果の全体像を理解することが可能となる。さらに本インタフェースを用いた提示法には、一次元リストでは膨大なリストに埋もれてしまい発見できない Web ページを発見することができるという利点もある。

また、検索結果が大量になると樹形図の各部分木が巨大になり、樹形図をそのまま表示するとユーザが情報を収集する際に情報が過多になることが考えられる。この問題を解決するために、各部分木の開閉をユーザが動的に行えるようにするような方法が考えられる。具体的には巨大な部分木のノードを省略し、ラベルのみを表示しておく。ユーザはラベルを参照しながら必要であると考えられる部分木を選別し、部分木を動的に展開することでノードを出現させ、必要な Web ページを得ることができるなどの手法がある。あるいは、Hyperbolic Tree の特性を利用し、中央に移動させた部分木は自動的に展開され、それ以外の部分木は閉じておくといった方法も考えられる。

6.1.2　ラベルの有用性

一般的なディレクトリ検索エンジンや、クラスタを動的に生成するインタフェースでは、カテゴリ間の関連が明示されていない。総合的な情報を必要とする情報指向型検索を行う際には各カテゴリ間の関連を考慮しながらユーザの思考を切り替えることなく情報をたどることができることが望ましい。

本インタフェースでは類似した部分木同士がラベルによって関連付けられているために、ユーザの思考を切り替えずに情報をたどる事が可能となっている。例えば風景に関する部分木から地理に関する部分木、地図に関する部分木と順に関連した部分木をたどって情報を収集することが可能である。

6.1.3 要旨の提示について

現在のシステムの実装では、ユーザが Web ページを表すノードを画面中央にドラッグしてクリックすることでシステムは Web ページの要旨を提示する。他の手法として、中央に移動されたノードは自動的に要旨を提示させるようにする方法が考えられる。これは Hyperbolic Tree の focus+context 手法をいかした要旨の提示法であるといえる。一次元リストの表示において、注視点を詳細に表示し、注視点から離れるほどユーザに与える情報を少なくする手法として bracketing が提案されている [23]。bracketing を二次元の Hyperbolic Tree に適応することでユーザはノードをクリックすることなく Web ページの要旨を閲覧することが可能となると考えられる。

しかし、この手法では注視点を移動させる際に自動的に要旨が表示されたり、ノードの数が多くなってしまった場合に検索結果提示画面が見にくくなってしまう可能性があるなどの問題が存在すると考えられる。

6.1.4 スクラップ機能の利点

本研究では Web ページの一部をスクラップブックに保存する機能を提供した。このように検索した Web ページの情報の一部を保存して、後に利用するためのアプリケーションとして、インターネットスクラップブック [24] や Google Notebook[25] などが存在する。本研究ではこれらのアプリケーションとは異なり、Web ページの一部を画像として保存する。このため、本システムにおいては重要な箇所をマーカーで強調したものを保存するなど、ユーザが切り取った Web ページの一部に任意の編集を行うことができるという利点がある。また、将来的にはそのようなユーザが編集した任意の記号を元に必要な Web ページの一部を検索するなどの応用が考えられる。

また類似アプリケーションはスクラップをフォルダに整理して保存したり、一次元のリストとして保存しユーザに提示するが、本研究のように二次元空間上の任意の場所にラベルをつけてスクラップを保存する手法は、他のアプリケーションに比べてスクラップが整理されている状態を視覚的理解することができるため、ユーザが後にスクラップブックを利用する際により便利であると考えられる。

6.2 処理時間について

本システムではまず GoogleAPI を用いて検索結果の URL を取得し、URL から HTML ファイルを得て分析し、クラスタリングを行っている。これらの処理を検索の度にすべて実行していては処理時間がかかりすぎるため現実的とはいえない。処理時間を短縮し、実用性のあるシステムにする必要がある。

クラスタリング処理については検索ごとに変化するが、HTML ファイルの分析結果は Web ページの内容が変化しない限り不変であるので、HTML ファイルの分析までを前処理として事前に処理しておくことで、全体の処理時間が短縮できると考えられる。

処理を行うタイミングは、一度得た URL の HTML ファイル分析結果をデータベースに蓄積しておき、以降の検索ではデータベースに登録されている URL であれば処理を行わずにデータベースから取り出すという方式が考えられる (図 6.1)。また、定期的に Web を巡回してデータベースに URL と HTML ファイルの分析結果を登録し、検索時に分析結果を利用するという方式も考えられる。前者の方式は一度目の検索に時間がかかり、後者の方式は Web の規模が膨大であるために分析結果を保存しておく記憶容量が膨大になるという問題点がある。これらの問題については今後検討しなければならない。

また、階層的クラスタリングの一般的な時間計算量は $O(n^3)$ であるが、これを改善する高速化アルゴリズムが提案されている [26][27]。また、文書ベクトルの次元数を圧縮することでクラスタリング計算時間を短縮することが可能である。

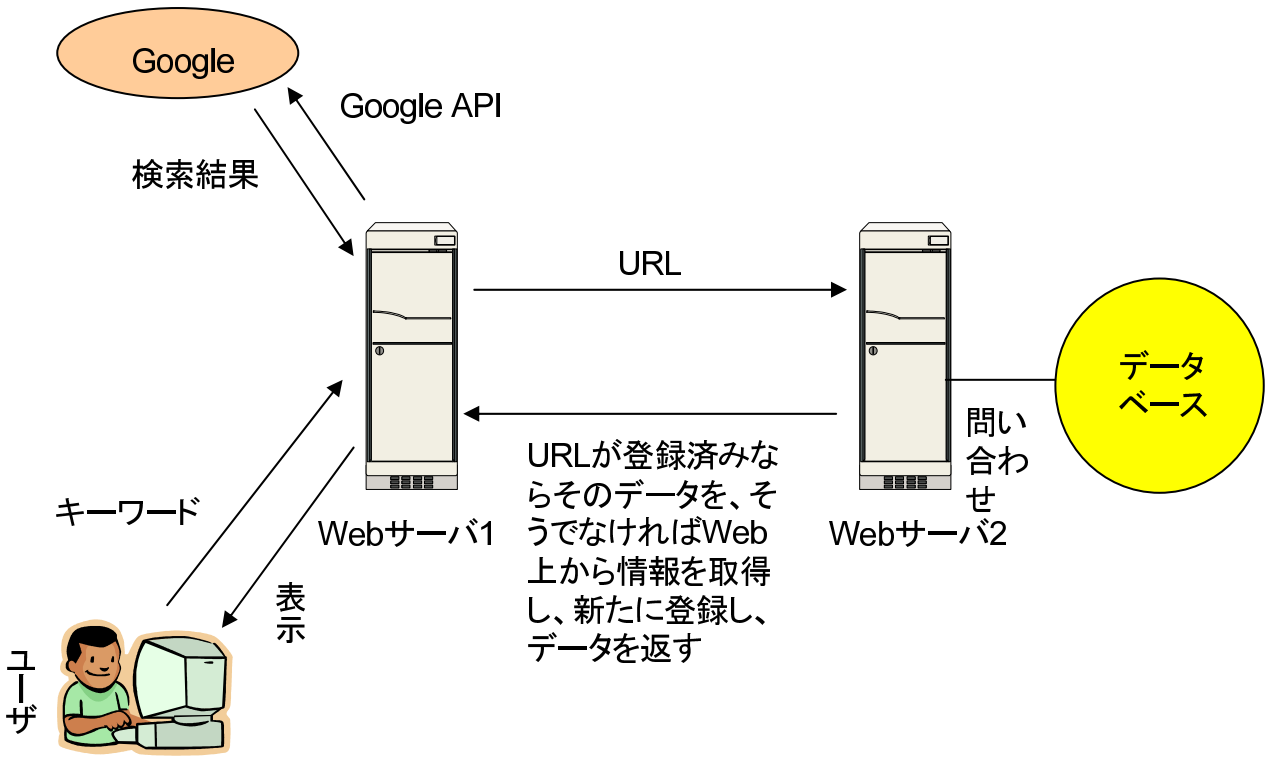

図 6.1: 処理時間の短縮

第7章　まとめ

本研究ではWebとWeb検索の現状について考察し、既存の検索インタフェースを用いて情報指向型検索を行う際の問題点を解決する決手法として概観提示インタフェースの提案と実装を行った。また、情報収集をより効率よく行い、情報検索と情報の保存を一連の作業として行うために概観提示インタフェースの機能拡張を行った。これによってユーザは情報指向型検索を行う具体的な場面において情報収集インタフェースとして本インタフェースの利用が可能となった。

謝辞

本研究を進めるにあたり、指導教官である田中二郎教授に様々な助言やご指導をいただきました。深く感謝いたします。また、志築文太郎講師にはグループミーティングなどにおきまして適切なご指導をいただきました。深く感謝いたします。有益な助言、ご指導を下さった三末和男助教授ならびに高橋伸講師に心から感謝いたします。また、田中研究室の皆様および WAVE グループの皆様には研究の全般にわたり丁寧なアドバイスをいただきました。本当にありがとうございました。最後に、私を支えてくださった家族や友人に改めて感謝いたします。

参考文献

[1] Inc. Internet Systems Consortium. http://www.isc.org/index.pl?/ops/ds/.

[2] Yahoo!Japan. http://www.yahoo.co.jp/.

[3] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 401–408. ACM Press/Addison-Wesley Publishing Co., 1995.

[4] Google. http://www.google.co.jp/.

[5] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, Vol. 36, No. 2, pp. 3–10, 2002.

[6] Barnard J.Jansen, Amanda Spink, and Tefko.Saracevic. Real users, and real needs:A study and analysis of usr queries on the web. *Information Processing and Management*, Vol. 36, No. 2, pp. 207–227, 2000.

[7] 原田昌紀, 佐藤進也, 風間一洋. 索引篩法 - 大規模サーチエンジンのための高速なランキング検索法. *DEWS*, 2003.

[8] Wojciech Wiza, Krzysztof Walczak, and Wojciech Cellary. Periscope: a system for adaptive 3D visualization of search results. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pp. 29–40. ACM Press, 2004.

[9] Jonathan Roberts, Nadia Boukhelifa, and Peter Rodgers. Multiform Glyph Based Web Search Result Visualization. the Sixth International Conference on Information Visualisation (IV ' 02), pp. 549–554. IEEE, 2002.

[10] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: detail and context smoothly integrated. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 173–176. ACM Press, 1991.

[11] Vivisimo. http://search.vivisimo.com/.

[12] Clusty. http://clusty.jp/.

[13] Kartoo. http://www.kartoo.com/en_index.htm/.

[14] 小林拓海, 佐藤大介, 三末和男, 田中二郎. Web 検索結果の概観提示による情報収集支援インタフェース. 人工知能学会第 19 回全国大会 CD-ROM, 2005.

[15] Takumi Kobayashi, Kazuo Misue, Buntaro Shizuki, and Jiro Tanaka. Information gathering support interface by the overview presentation of web search results. In *Proceedings of Asia Pacific Symposium on Information Visualization 2006 (APVIS2006)*, pp. 103–108, Tokyo, Japan, Feb 2006.

[16] Brain S.Everitt. *Cluster analysis*. London:E.Arnold, 3rd edition, 1993.

[17] 形態素解析・構文解析入門. http://www.unixuser.org/ euske/doc/nlpintro/.

[18] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 16–23, New York, NY, USA, 1986. ACM Press.

[19] Manojit Sarkar and Marc H. Brown. Graphical fisheye views. *Commun. ACM*, Vol. 37, No. 12, pp. 73–83, 1994.

[20] Google Web APIs. http://www.google.com/apis/.

[21] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸. 形態素解析システム「茶筅」version 2.2.1 使用説明書. Technical report, 奈良先端科学技術大学院大学 松本研究室, 2000.

[22] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[23] Jonathan C Roberts and Edward Suvanaphen. Visual bracketing for web search result visualization. In Ebad Banissi et al, editor, *Proceedings Information Visualization (IV03)*, pp. 264–269. IEEE Computer Society, July 2003.

[24] Atsushi Sugiura and Yoshiyuki Koseki. Internet scrapbook: automating web browsing tasks by demonstration. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 9–18, New York, NY, USA, 1998. ACM Press.

[25] Google notebook. http://www.google.com/notebook/.

[26] Sung Jung and Taek-Soo Kim. An agglomerative hierarchical clustering using partial maximum array and incremental similarity computation method. In *ICDM*, pp. 265–272, 2001.

[27] 石橋徹夫, 古賀久志, 渡辺俊典. Locality-sensitive hashing を用いた階層的クラスタリング手法. 電子情報通信学会論文誌, 第 J88-D-Ⅱ, NO4 巻, pp. 852–863, 2005.