平成16年度

筑波大学第三学群情報学類

卒業研究論文

題目 Web検索結果の概観提示による情報収集支援システム

主専攻　　情報科学主専攻

著者　　小林 拓海

指導教員　田中二郎、志築文太郎

要　旨

現在インターネット上に存在する Web ページの数は数十億以上にのぼる。そのような膨大な数の Web ページの中から必要な情報を取捨選択することは容易ではなく、一般的に利用されている既存の検索エンジンのように、ジャンルを問わず検索結果を数十ページに分割し、テキストのリストで提示するようなインタフェースは必ずしもユーザにとって使い勝手が良いとは言えない。

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、それらを解決する手段として、Web 検索結果を分析し、内容に応じて分類することで検索結果全体としての特徴を明らかにする。さらに、特徴の直観的な理解を支援するために、分類した Web ページを 1 画面に納めて適切に配置することで Web 検索結果の概観をユーザに提示するシステムを提案し、実装を行った。本システムを用いることでユーザは Web 検索結果の概観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。

目 次

第 1 章　序論 **1**

第 2 章　Web 検索の現状と問題点 **3**
2.1 Web 検索の現状と考察 3
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 5

第 3 章　概観提示システム **6**
3.1 概観提示システムのための提案 6
3.2 提案手法と他の類似インタフェースとの相違点 6
3.3 本システムで利用する要素技術 8
3.3.1 クラスタリング 8
クラスタリングとは 8
形態素解析 9
tf/idf 法 10
ベクトル空間法 10
3.3.2 クラスタリング結果の可視化法 10

第 4 章　システムの流れと詳細 **13**
4.1 検索結果の分析とクラスタリング部 13
4.1.1 検索質問の入力と検索結果の取得 13
4.1.2 HTML ファイルへの変換 13
4.1.3 HTML ファイルの分析 13
4.1.4 クラスタリング 14
4.1.5 Web 文書クラスタリング結果の考察 15
4.2 概観提示部 15
4.2.1 概観提示画面 15
4.2.2 クラスタリングにおける問題点の改善 18
4.3 システムの相互関係 20

第 5 章　システムの実用例 **22**

第 6 章　関連研究 **26**

第 7 章　結論 **27**

謝辞 **28**

参考文献 **29**

図 目 次

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] 1

2.1 Google の検索結果提示画面 4

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) 7
3.2 Hyperbolic Tree(John Lamping)[8] 11
3.3 Hyperbolic Tree のフォーカスの移動 12

4.1 「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 . 16
4.2 システムの提示画面 17
4.3 Web ページ A と Web ページ B の関係 18
4.4 クラスタリングが有効に行われている部分木 19
4.5 表示インタフェース的なクラスタリングの改善 19
4.6 システムの流れと相互関係 20

5.1 検索クエリ「日本　歴史」に対する提示画面 23
5.2 地理に関する部分木 24
5.3 教科書に関する部分木 24
5.4 学会や論文に関する部分木 25

6.1 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 26

表 目 次

3.1 本研究とディレクトリ型検索との相違点 8
3.2 「私は筑波大学に通っています。」を形態素解析した例 9

4.1 HTML 文書におけるタグの例 13

第1章　序論

現在、インターネットはより一般的なものとなり、情報を収集する場面など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[2] は 2005 年現在約 80 億もの Web ページから検索を行っている。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、2004 年 7 月現在のインターネットに接続されたホストの数は約 2 億 8 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。

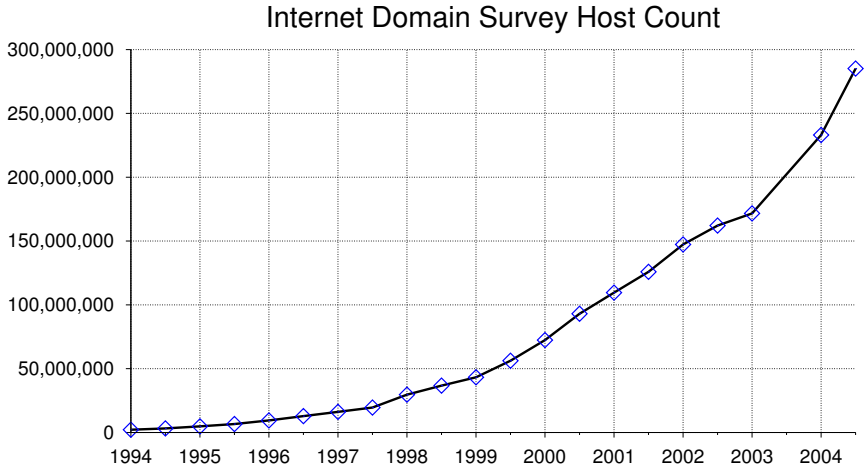

図 1.1: インターネットに接続されているホスト数の推移 (Internet Systems Consortium)[1]

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など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。

本研究の目的

本研究の目的は Web 検索の現状を理解し、その問題点を解決するシステムを考案し、実装することである。そのために本研究では 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手法および分類手法を提案し、検索結果全体の特徴を明確にする。さらに、ユーザの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、分類した Web ページを 1 画面に納めてユーザに提示する手法を提案した。

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 検索および検索インタフェースについて考察し、その問題点を既存のインタフェースと比較しながら述べる。第 3 章では問題解決手法として概観提示システムを提案し、その特徴と利用する要素技術について説明する。第 4 章ではシステムの流れと詳細について説明し、第 5 章ではシステムを用いた検索の実例を挙げる。第 6 章では関連研究について触れ、第 7 章で結論を述べる

第2章　Web検索の現状と問題点

本章ではまず日常のWeb検索の現状を考察し、さらにその問題点について述べることで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[3]。

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。

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 [4]。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[1]において、1ヶ月に用いられた検索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 [5]。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。

[1]http://odin.ingrid.org/　　現在はサービスを休止している。

以上のようなWeb検索の現状を考慮し、一般的な検索エンジンが提示する検索結果のように、検索エンジンが重要であると判断したWebページから順にジャンルを問わずにユーザに提示するインタフェースは不十分であると考えた。

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

情報指向型の検索を行う場合、ユーザは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。

第3章　概観提示システム

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問題点を改善する手法として概観提示システムを提案し、概観提示システムを実現するために 2 つの目標を掲げた。次に、提案システムと他の類似インタフェースとの相違点を述べ、さらに提案システムに用いる要素技術を各々簡単に説明する。

3.1　概観提示システムのための提案

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問題点を改善する方法として、概観提示システムを提案する。提案したシステムを実現するために、以下の 2 点の目標を考えた。

目標 1 : Web 検索エンジンによって与えられる検索結果のクラスタリング (分類)

目標 2 : クラスタリング結果を利用して Web 検索結果を 1 画面に納めてユーザに提示

目標 1 は、Web 検索エンジンによって与えられる検索結果である複数の Web ページを分析し、それぞれの類似度によって適切にクラスタ (分類された Web ページ群) を作成し、各クラスタの特徴を表すラベルを付加することを目的とする。

目標 2 は、目標 1 によってクラスタリングされた Web 検索結果を概観が理解しやすいように 1 画面に納まるようにユーザに提示することを目的とする。本研究ではユーザへの提示に “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

ここでは提案システム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、本システム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。

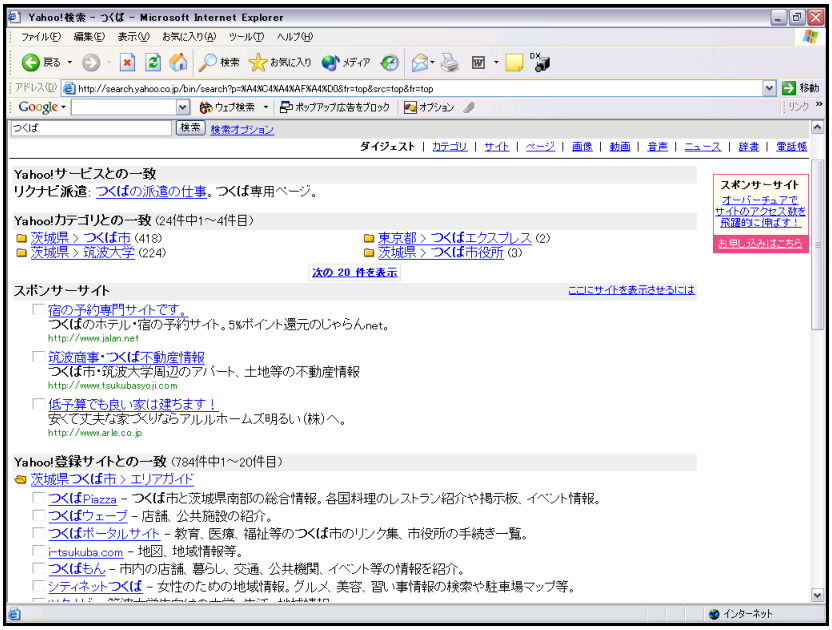

図 3.1: ディレクトリ型検索エンジンの検索結果提示画面 (Yahoo!Japan)

「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。

ここで本研究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、本研究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、本研究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、本研究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、本研究においては Web ページのデータをあらかじ

表 3.1: 本研究とディレクトリ型検索との相違点

	ディレクトリ検索	本研究
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	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

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、本研究では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、本研究では 1 画面に納める事が可能となっている。

3.3 本システム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、概観を表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 [6]。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 [7]。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。

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

[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

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。

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

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

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[8] らによって提唱された、双曲空間上に表現された樹形図である。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、中央に近いノードほど大きく表示され、中央から離れたノードほど

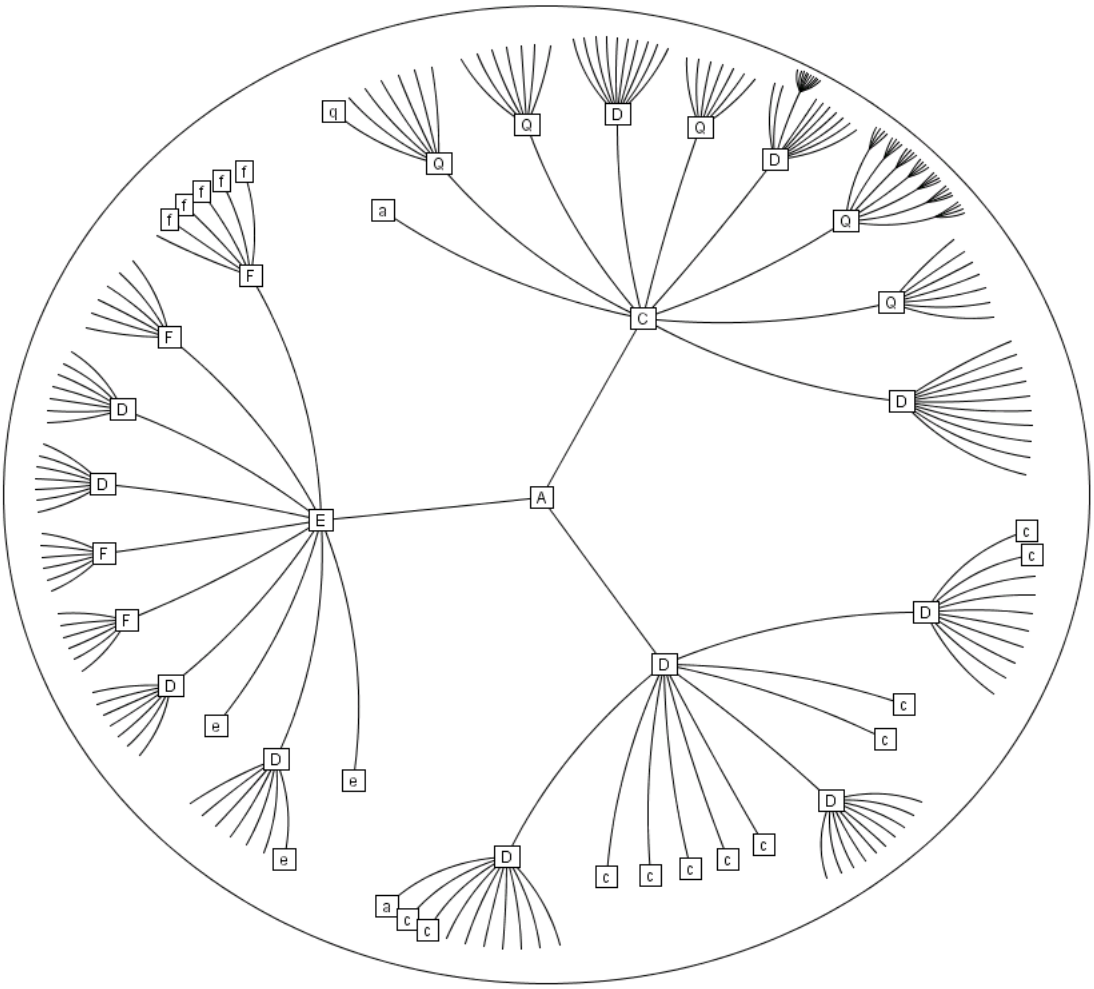

図 3.2: Hyperbolic Tree(John Lamping)[8]

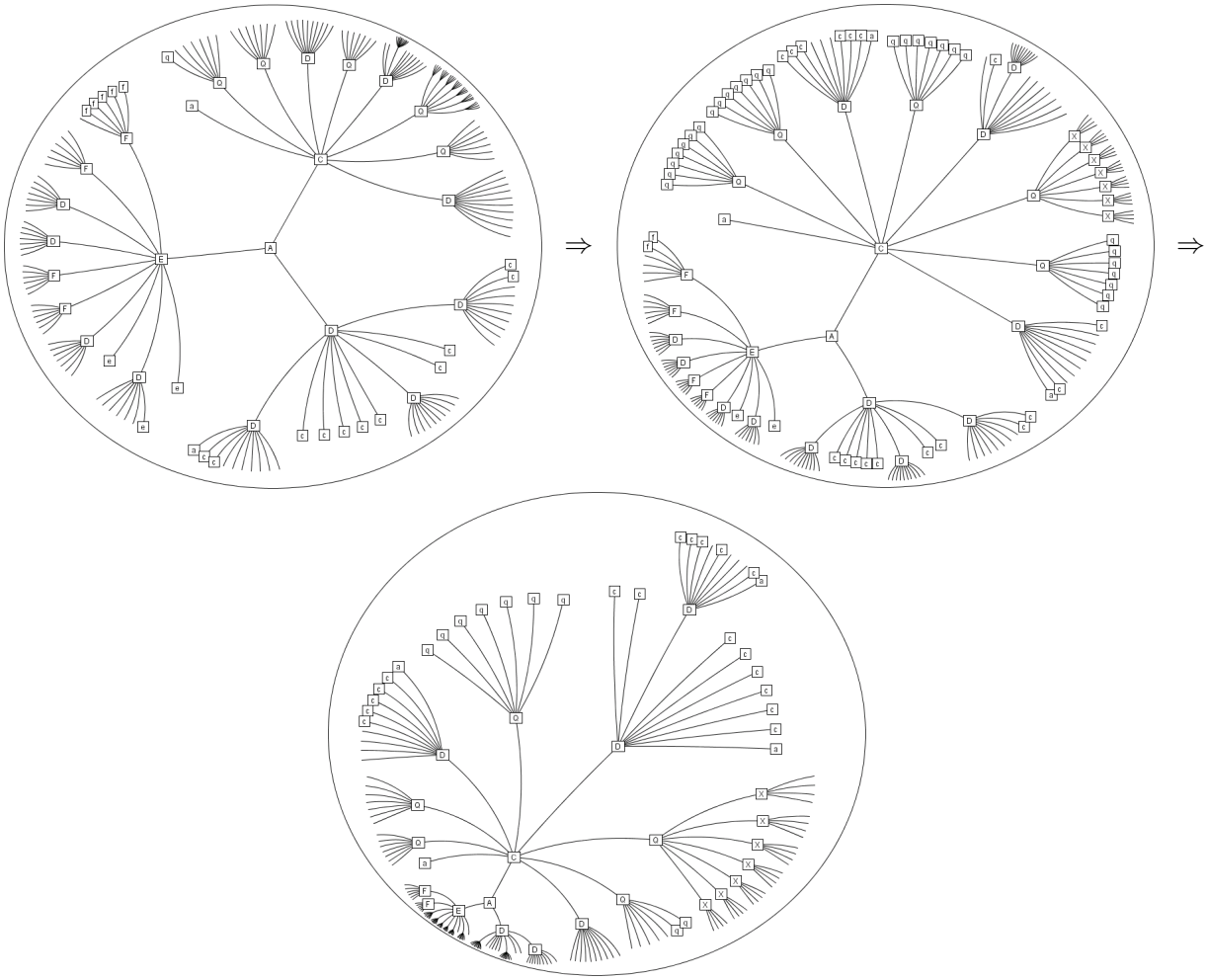

図 3.3: Hyperbolic Tree のフォーカスの移動

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離れたノードが中央に近づき、小さく表示されていたノードが大きくなる。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。

第4章　システムの流れと詳細

4.1　検索結果の分析とクラスタリング部

4.1.1　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[9] 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。

4.1.2　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。

4.1.3　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 4.1に HTML 文書に現れるタグの一例を挙げた。

表 4.1: 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。

4.1.4 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 [10]。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、

$$tf_{ij} = \log(1 + f_{ij}) \tag{4.1}$$

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、

$$idf_j = \log(\frac{n}{d_j}) \tag{4.2}$$

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。

$$n_i = \sqrt{\sum_{j=1}^{m}(tf_{ij} \times idf_j)^2} \tag{4.3}$$

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。

$$w_{ij} = \frac{tf_{ij} \times idf_j}{n_i} \tag{4.4}$$

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。

4.1.5 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 4.1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。

4.2 概観提示部

本システムでは、ユーザの入力した検索クエリに対しての検索結果をクラスタリングし、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。

4.2.1 概観提示画面

図 4.2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。各ノード間をつなぐエッジは中央ほど長く、中央から離れるほど短く表示される。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。

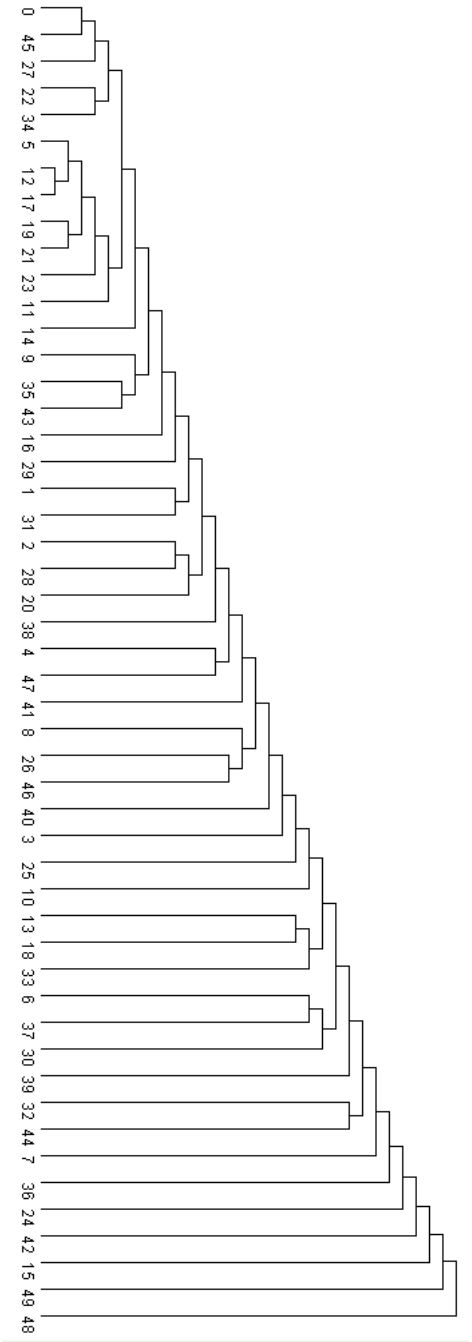

図 4.1: 「小林」という検索クエリで 50 件の Web ページをクラスタリングした結果

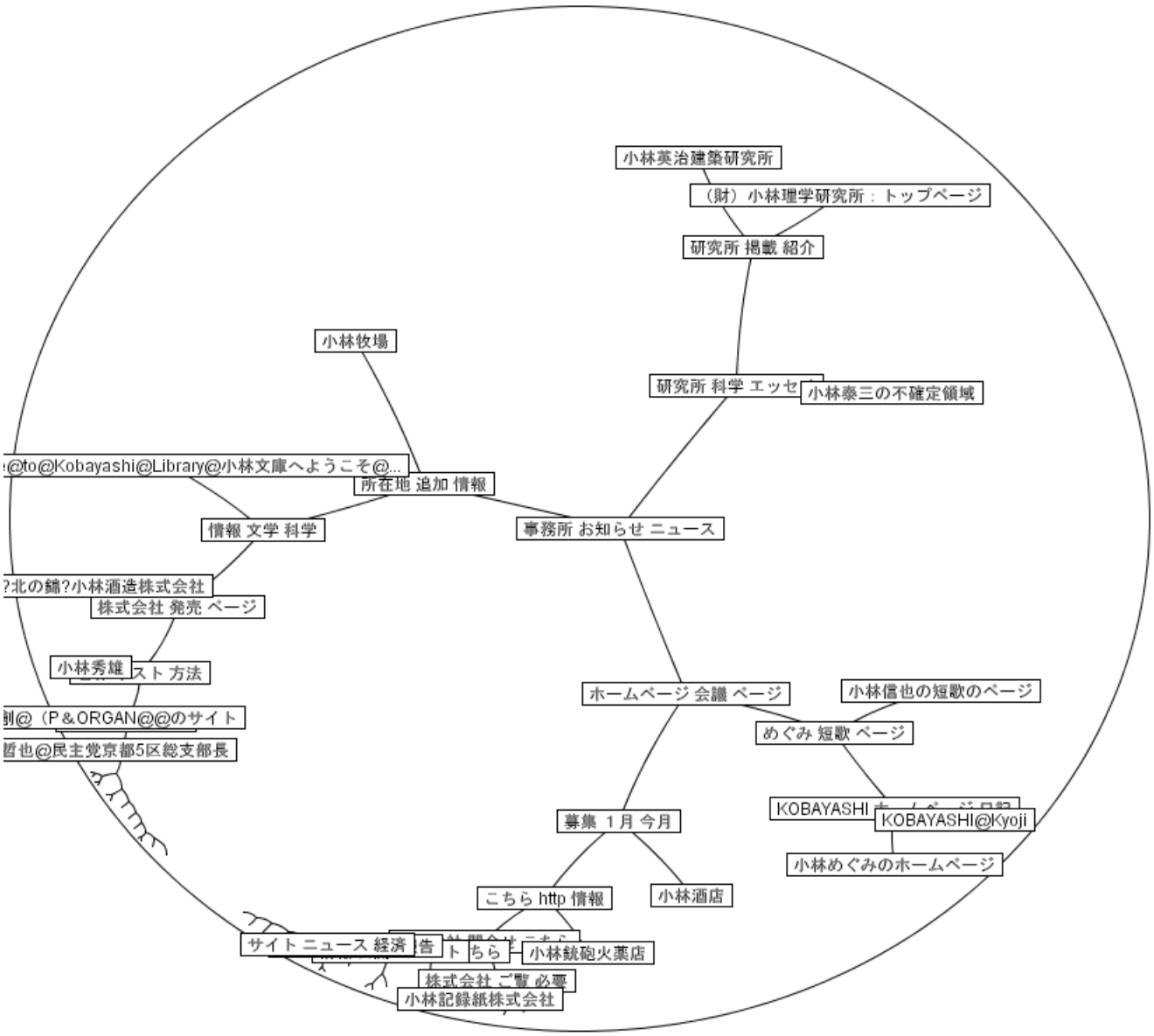

図 4.2: システムの提示画面

次に各ノードの関係について述べる。親ノード単語 1 単語 2 単語 3 と子ノード Web ページ A と子ノード Web ページ B が図 4.3のように接続されている場合を考える。

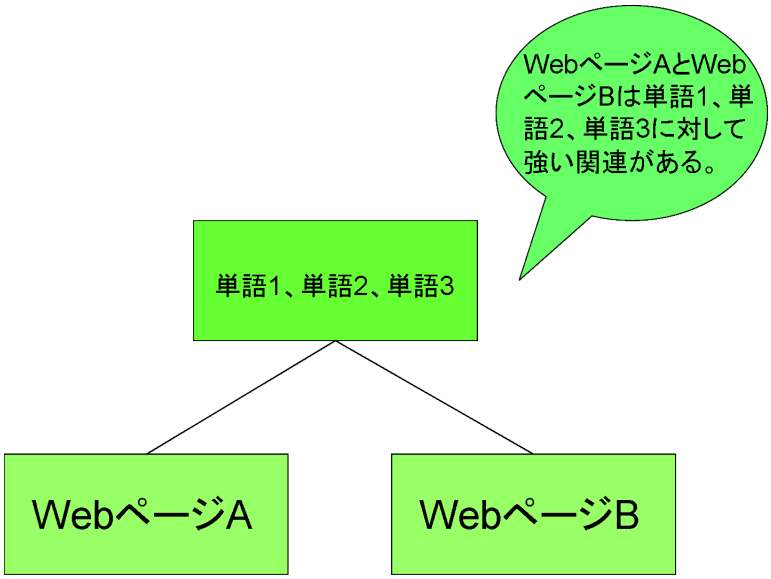

図 4.3: Web ページ A と Web ページ B の関係

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。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 4.4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。

4.2.2　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 4.5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

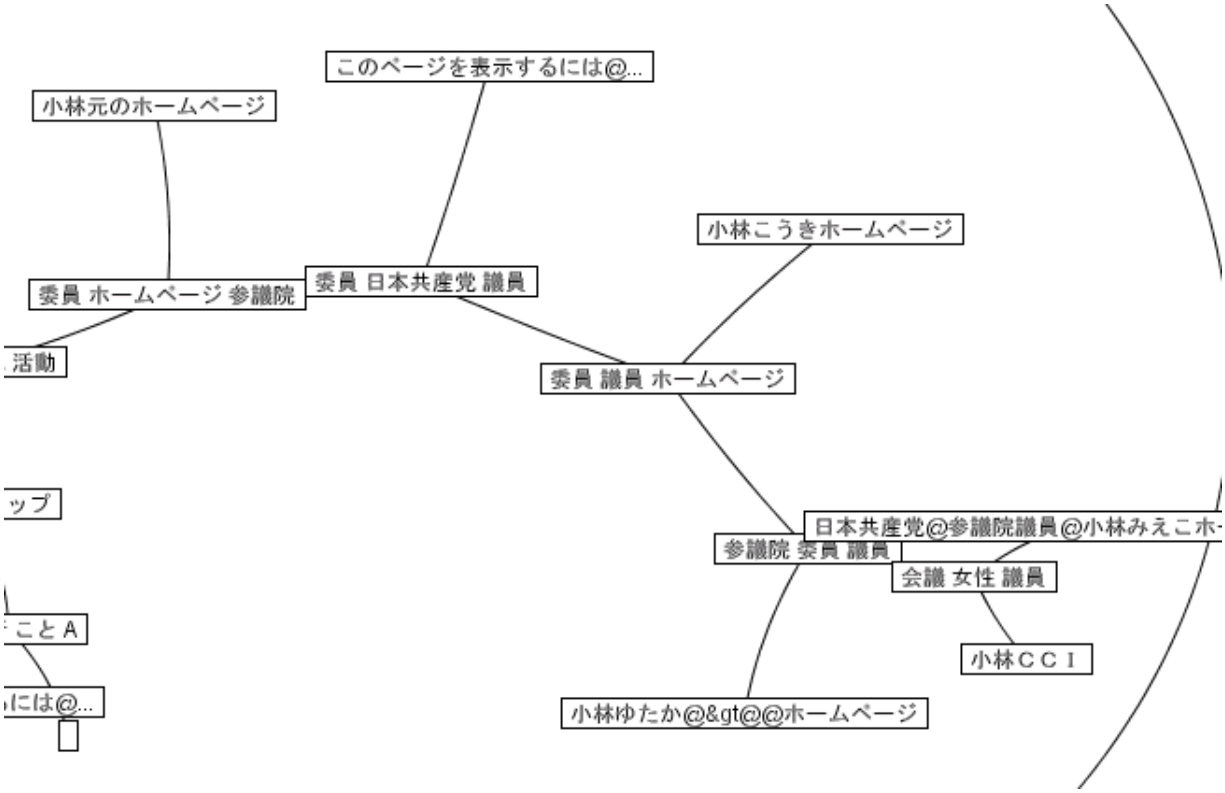

図 4.4: クラスタリングが有効に行われている部分木

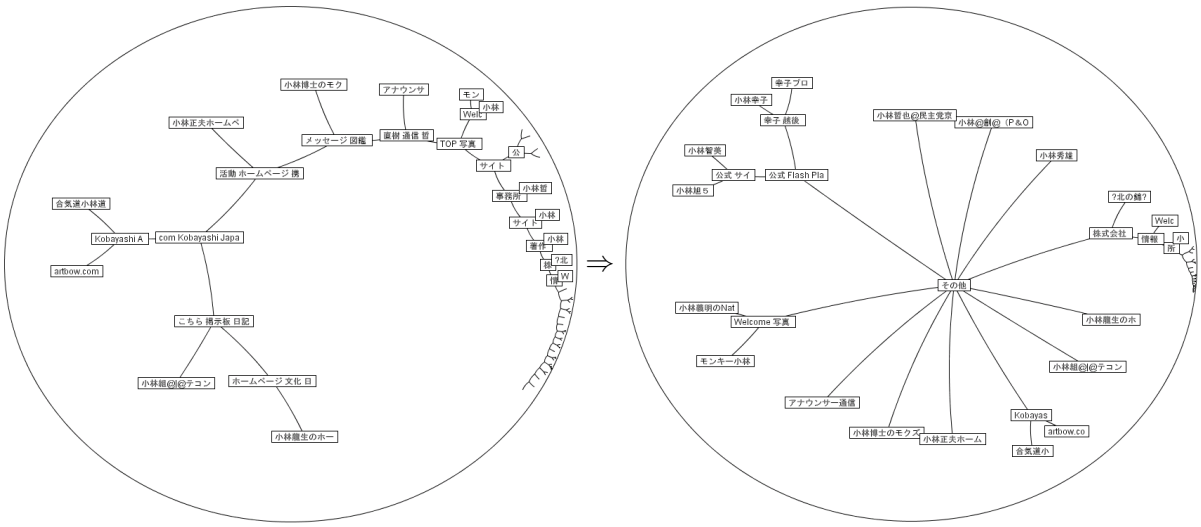

図 4.5: 表示インタフェース的なクラスタリングの改善

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。

4.3 システムの相互関係

ここでシステムの流れと相互関係を図 4.6にまとめておく。

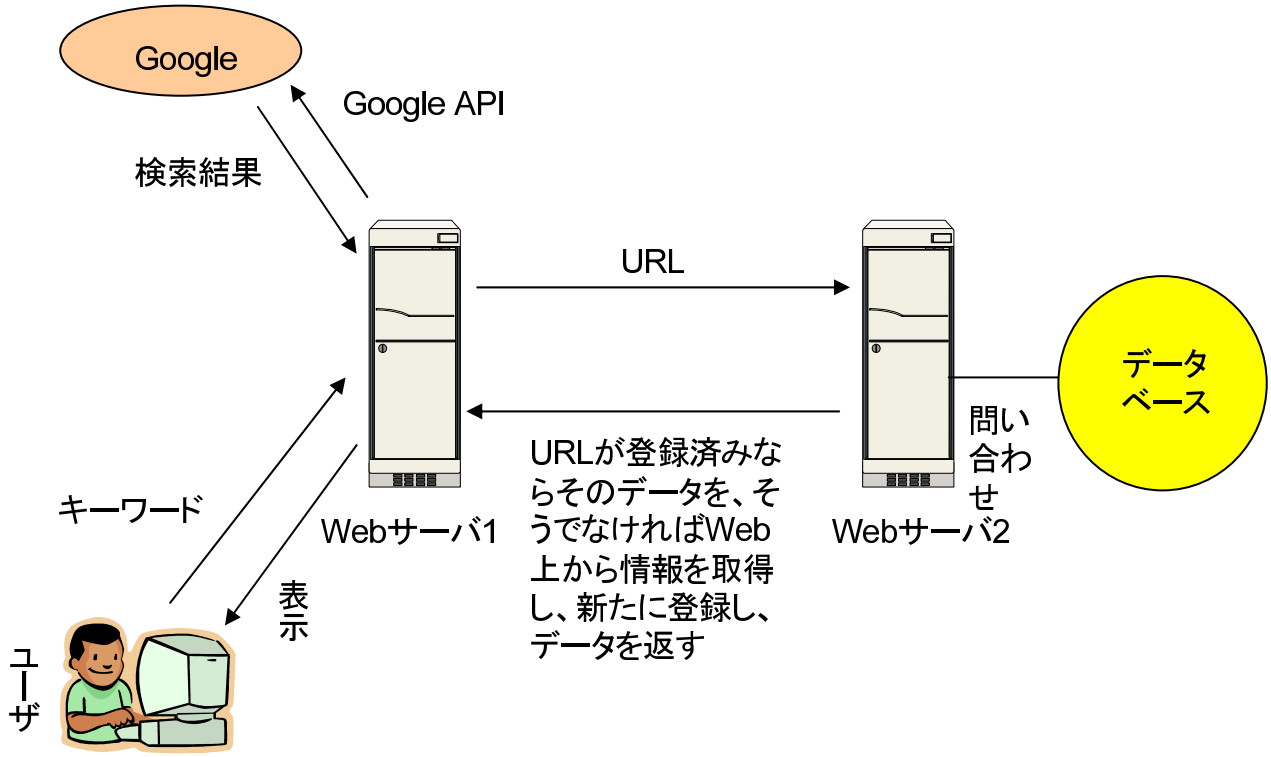

図 4.6: システムの流れと相互関係

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。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。

第5章　システムの実用例

ここでは具体的な検索場面の例を挙げて本研究のシステムを適用する。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いと思い、本システムに検索クエリ「日本　歴史」を与えた。システムはアルゴリズムにしたがってクラスタリングし、結果をユーザに提示する。提示された情報の中からユーザは必要な情報を取捨選択する。

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。本研究のシステム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 5.2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 5.3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べているページが多いことが分かる。図 5.4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。

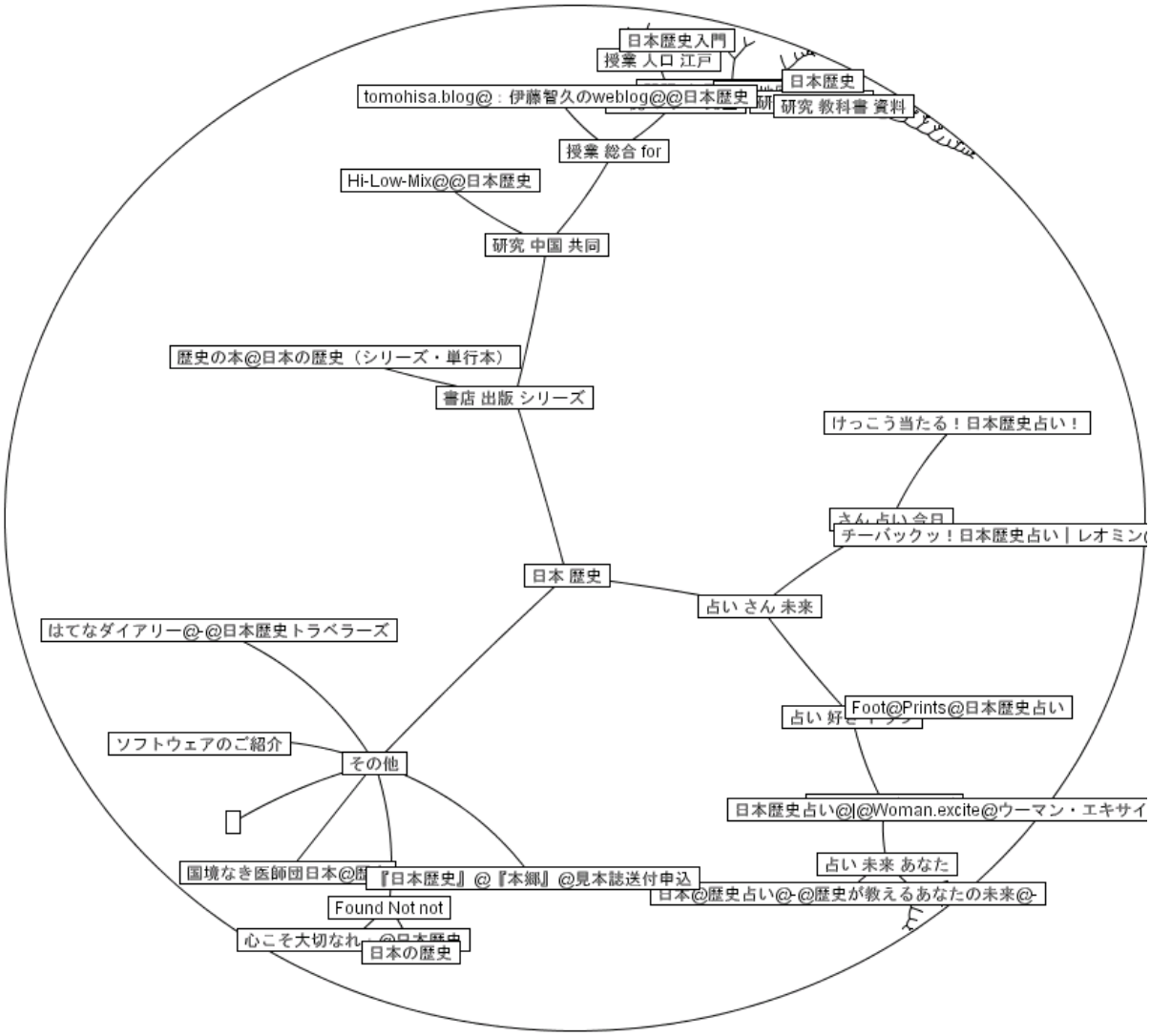

図 5.1: 検索クエリ「日本　歴史」に対する提示画面

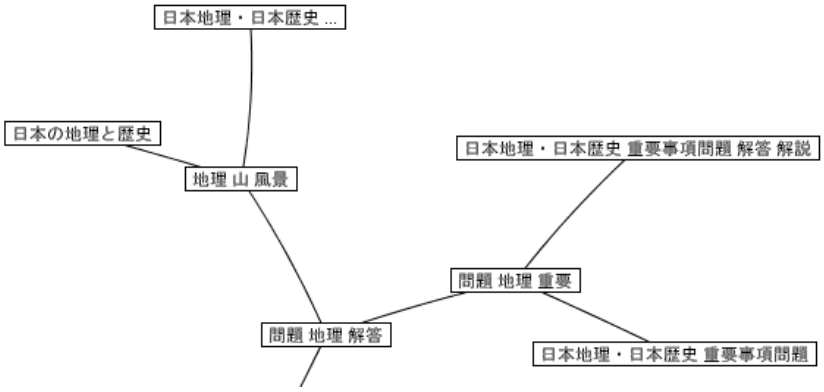

図 5.2: 地理に関する部分木

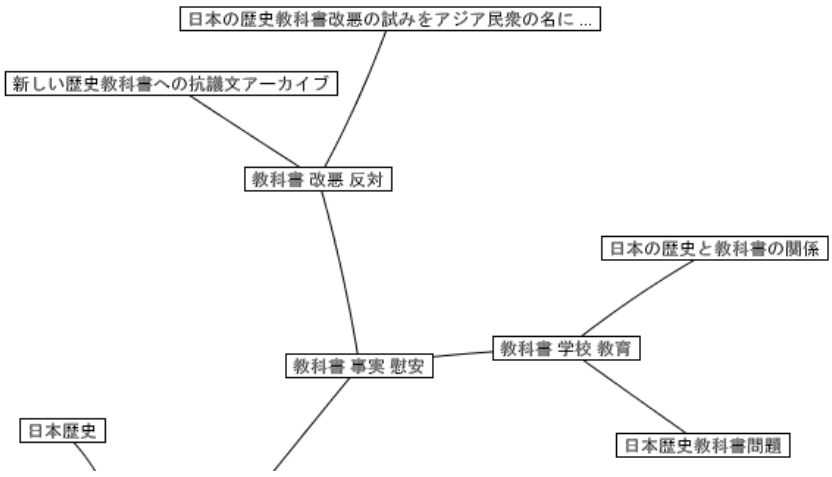

図 5.3: 教科書に関する部分木

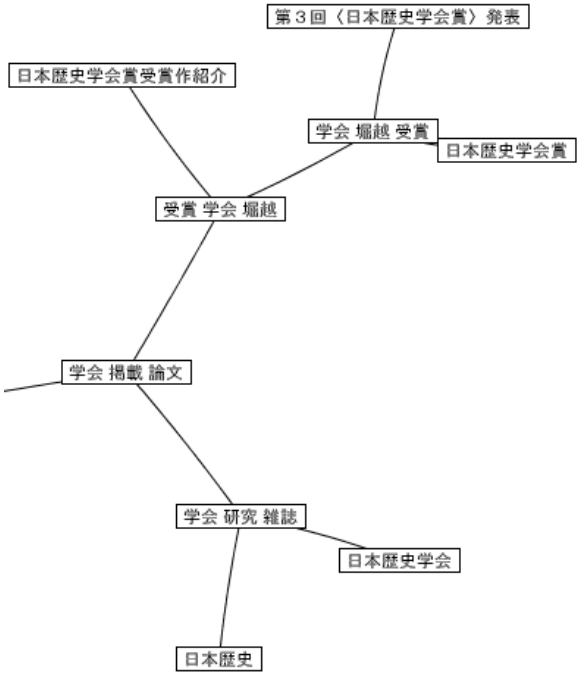

図 5.4: 学会や論文に関する部分木

第6章　関連研究

Poznan University of Economics, Department of Information Technologyの開発するPeriscope[11]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 [12]。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。

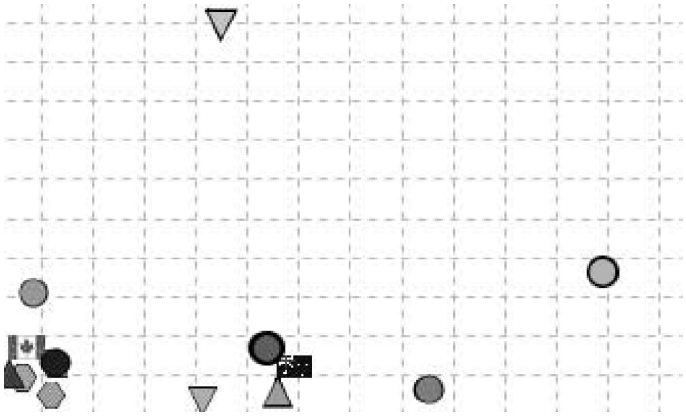

図 6.1: Webページをホスト名でクラスタリングし二次元空間に配置した例

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[13]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。

第7章　結論

本稿ではまず現在の Web と Web 検索の現状を考察し、既存の検索インタフェースの問題点を述べた。次に問題解決の手法として概観提示システムの提案と実装を行い、システムの活用例を述べた。本システムによってユーザの Web 検索結果の概観理解および情報収集を支援することが可能となる。

今後は提示インタフェースの改良や検索時間の短縮などを行い、ユーザにとってさらに使いやすいシステムに改良すること、さらにシステムのユーザ評価を行って開発に役立てたいと考えている。

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、三末和男助教授にはチーム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、ご指導を下さった志築文太郎講師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チームリーダー佐藤大介さんをはじめとする SWAT チーム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。

参考文献

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

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

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

[4] 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.

[5] 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.

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

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.