

階層的クラスタリングを利用したメタ検索エンジンの提案 ~ METAL ~

成田 宏和 太田 学 片山 薫 石川 博
東京都立大学 大学院 工学研究科

E-mail: h-narita@jcom.home.ne.jp, {ohta,katayama,ishikawa}@eei.metro-u.ac.jp

Web 上で必要な情報へのアクセス手段として、中心的な役割を担っているのが google, goo, Lycos などに代表されるロボット型 Web 検索エンジンである。これらの検索エンジンが日々変化し続ける Web に対応していく為には、新しいページの収集、更新チェックなど多大なメンテナンスを必要とする。一方で、検索結果があまりに膨大すぎて必要な情報を探すのに困難を伴う場合もあるし、全ての検索エンジンが同一の情報を収集しているわけではないので、或る検索エンジンでは存在しなかった情報も別の検索エンジンを利用すると発見できる場合もある。本稿では、検索能力の強化と発見しにくい情報へのアクセス支援を目的に、複数の検索エンジンを併用し、検索結果を階層的にクラスタリングしてユーザに提供するシステムである METAL を提案する。

- メタ検索エンジン, 階層的クラスタリング, 形態素解析, 特徴語, 茶筌 -

Meta Search Engine Using Hierarchical Clustering - METAL -

Hirokazu Narita Manabu Ohta Kaoru Katayama Hiroshi Ishikawa
Graduate School of Engineering, Tokyo Metropolitan University

E-mail: h-narita@jcom.home.ne.jp, {ohta,katayama,ishikawa}@eei.metro-u.ac.jp

It is search engines represented by google, goo, Lycos, and so on that shoulders a leading role as an access means to the necessary information on the Web. A lot of maintenance, such as collection of new Web pages and renewal check, is necessary for these search engines to cope with the Web changing continually. On the other hand, sometimes, it is difficult to find necessary information because of enormous search results. One search engine may be able to find what another can't because different search engines collect different kinds of information. In this paper, we propose a meta search engine which uses more than one search engine at the same time for improving recall, and cluster search results for supporting users to find their necessary information more easily.

- Meta search engine, hierarchical clustering, morphological analysis, feature word, ChaSen -

1. はじめに

1993年にJPNICが設立され、日本のインターネット業界は急速に発展をしてきた。1993年当初は、新着情報やディレクトリなどの検索サイトが貴重な存在であった。Webの情報量が増えるにつれて、人間の編集者によって分類されたディレクトリを辿るディレクトリ型だけではなく、Webロボットを利用して収集したWebページをキーワードで検索するロボット型検索エンジンも登場するようになった。第一世代のロボット型検索エンジンは、ブラウザに表示されるテキストのみなど、コンテンツの一部分しか処理できないものであった。次いで、コン

テンツ内のテキスト全てを検索対象とした全文検索システムが登場する。更に、タグを解析することで、逆リンク構造を反映した検索が出来る第二世代の検索エンジンが登場した。1995年12月にAltaVista[1]によって始められた、タグを含めて収集したページの内容を全てデータベース化した全文検索システムがそれである。しかしながら、日本語は単語の区切りが難しいため、英文全文検索システムに遅れて1997年3月末にHotBot[2]と提携したgoo[3]が日本語全文検索をはじめた。そして、今日のコンテンツ間の関連を考慮した第三世代の検索エンジンに至る。1999年10月にGoogle[4]

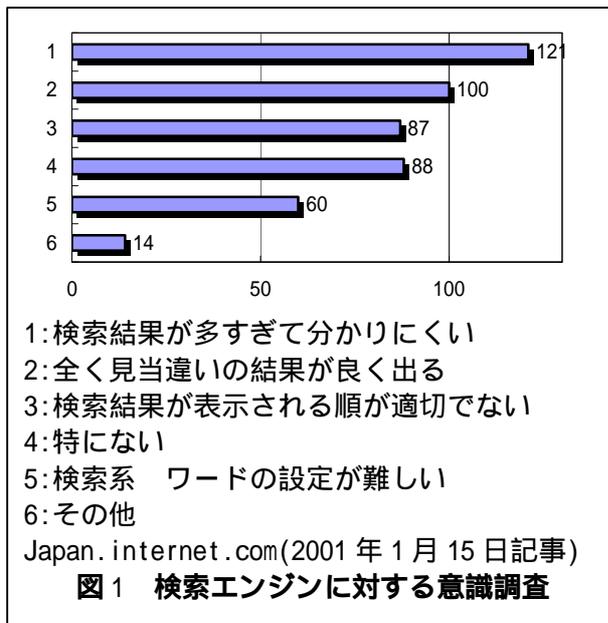
がリンク情報を利用した検索システムを登場させ、2000年9月から日本語検索サービスを本格的に始めた。

ページ間の関連を無視したページ単位の検索から、関連性を考慮した検索システムへと変遷し、関連情報も一緒に検索されるようになった。そのため、検索キーワードが存在しない Web ページも検索結果として出て来る場合がある。加えて、厳密なキーワードを用いなくても、同義語をある程度包含するようになった。

このような発展を遂げてきた検索エンジンだが、まだまだ十分とは言えない。図 1, 2 は Japan.internet.com[5]が行った検索エンジンのユーザ 300 人に対する調査である。図 1 は複数回答可能による調査で、検索エンジンに対し、検索結果が多すぎて分かりにくい、全く見当違いの結果が良く出るなどの不満を抱くユーザが、相当数いることが分かる。また、図 2 より

ある情報を探す際に、常にではないにせよ、複数の検索エンジンを利用するユーザは 9 割に上ることが分かる。この解決策として、複数の検索エンジンを効率的に利用するシステムとして、メタ検索エンジンがある。一回の検索で複数の検索エンジンに対して検索を行うシステムで、検索結果をそのまま表示する方式、統合して表示する方式、更にクラスタリングして表示する方式などがある。複数の検索エンジンから情報を収集することで、一つの検索エンジンでは得られなかった新たな検索結果を得ることが出来たり、ランキングを平均化することも可能である。

本稿では、メタ検索エンジンを採用し、雑多で分かりにくい検索結果から、目的の情報を発見する支援、見当違いの検索結果の分別を行うために、検索結果を統合しクラスタリングして表示するメタ検索エンジン METAL を提案する。



2. 関連研究

2.1. Vivisimo

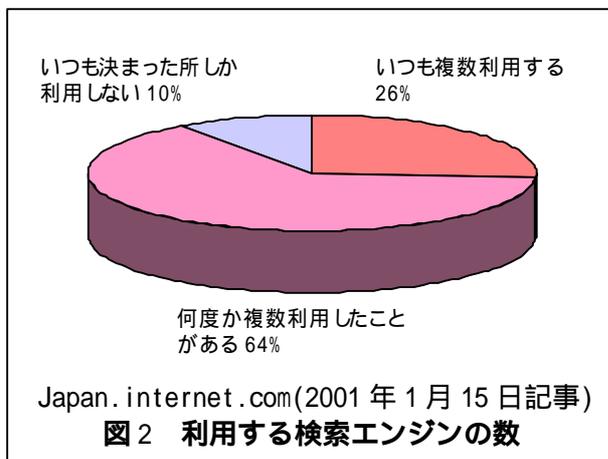
メタ検索エンジンの一種で、検索結果を階層的にクラスタリングして、クラスタをツリー型メニューとして表示する。クラスタリングには、Document clustering[6]と呼ばれる技術を利用しており、文書を意味のあるグループに自動組織化するものである。クラスタリングに利用する情報は、問い合わせを行った検索エンジンの出力結果から抽出したタイトル、サマリ、URL である。

2.2. ナレッジクエリサーチ (KQS)

Knowledge Query Server[7]を利用したメタ検索エンジンで、非階層クラスタリングを行う。各検索エンジンからの最大取得件数は、最高で 25 件まで設定できる。クラスタ数は、4~8 個程度であり、検索結果から得られたタイトル、サマリの単語を利用したり、URL からサイトを決定するなどしてクラスタリングしている。

2.3. Lycos

メタ検索エンジンではないが 2001 年 12 月に WiseNut の patent-pending context sensitive ranking system[8]を導入し、文脈を考慮したランキング、検索結果の自動クラスタリングシステム[9]を提供する。クラスタリングされる検索結果は、上位 100 件程度で、第一階層のクラスタ数は 4~8 個程度である。



3. メタ検索

以下で、METAL で利用している問い合わせ方法、検索結果の取得方法、検索結果の統合方法、ランキング方法について述べる。

3.1. 問い合わせ方法

Metcha Search[10]と同様に URL に環境変数として、検索キーワード、ページ内最大表示件数等を付与し HTTP プロトコルを用いてクエリを送信する。以下に goo の場合の例を図3に示す。

`http://goo.ne.jp/default.asp?MT=meta+searchengine&DC=50`

MT=に続けて検索キーワードを+で区切って付与
DC=に続いて1ページの最大表示件数を付与

図3 問い合わせ方法

3.2. 検索結果の取得方法

検索結果の HTML ソースを取得し、正規表現により、タイトル、サマリ、URL の取得を試みる。

3.3. 検索結果の統合方法

メタ検索を行うと、いくつかの検索エンジンが同じページを示す検索結果を返してることがある。この場合、これらの検索結果が別の物として扱われることを防止するために、次の条件を満たす URL を一つにまとめる。

- ・ 完全一致
- ・ 末尾の 『/』 を排除すると一致
- ・ 『http://』 後の 『www.』 を排除すると一致

例えば、以下の ~ は全て と同一の検索結果としてまとめる。

```
http://goo.ne.jp
http://goo.ne.jp/
http://www.goo.ne.jp
http://www.goo.ne.jp/
```

また、統合する際、検索エンジンによって表示される内容が異なる場合がある。検索デスク[11]によれば、本研究でメタ検索に利用している検索エンジンでは、表1のようにサマリを選択している。

表1 サマリの選択方法

goo [3]	descriptionを使用 ないときは段落前の文章から100字
Google [4]	検索キーワードを含む文章を100字 そのURLがgoogleのディレクトリに登録 されていればその抄録・カテゴリ
Infoseek [12]	descriptionを使用 ないときはタグを除いた最初から100字
TOCC [13]	descriptionを使用 ないときはタグを除いた最初から100字

description とは、サイト作成者により、HTML 内にメタタグによって記述されたサイトの概要である。

また、検索エンジンによっては文字化けしていることもある。文字化けした文章を表示している検索エンジンは、文字化けしていない検索エンジンよりも長い文章を表示している傾向があるので文字数の少ないものを統合した結果に利用している。

3.4. 検索結果のランキング

各検索エンジンでは、検索対象となる母集団もランキングシステムも異なるため、出力されるデータも表示順序も一致することは考えにくい。そこで、各検索エンジンでの順位を元に得点 p_e を求め、 p_e を利用して算出した総合得点を基に、統合した検索結果をランキングする。総合得点として、単純に p_e の平均とした場合不都合が生じる。メタ検索に用いている検索エンジンの数がそれほど多くないので、一つの検索エンジンの得点が平均点に大きく関与し、順位が大きく動いてしまう。そこで、一つの検索エンジンの影響で順位が大きく動くことを防止するために、 p_e と出現した検索エンジンの数を考慮して総合得点を決定する。まず、検索エンジン e での得点を 式により定義する。ここで、 max は e での最大取得件数、 n は e での順位である。

$$p_e = 100 \times \frac{max - (n - 1)}{max} \dots$$

次に、各検索エンジンからの得点を統合する方法を考える。一つの検索エンジンの影響でランキングが大きく変動することを防止するために、同一の Web ページを示す検索結果が 3 つ以上の検索エンジンから得られた場合、 p_e の最大値と最小値を省いて平均を採る。2 つ以下の場合はそのまま平均を取る。これをまとめ

ると、次の通りである。

- ・ 出現した検索エンジン数が 2 以下の場合
単純平均
- ・ 出現した検索エンジン数が 3 以上の場合
最大値と最小値を除いて平均を取る。また、
最大値、最小値が重複している場合は、各
一つずつを除く。

例えば、4 つの検索エンジンが同じ Web ページを示す検索結果を返し、 p_e が {100, 80, 60, 60} の場合、出現した検索エンジン数が 4 なので、最大値と最小値を除いて平均を取る。最大値は 100、最小値は 60 で重複しているので、一つだけ除く。よって、考慮するのは {80, 60} であり先述の定義に基づき計算すると 式のようになる。

$$point' = \frac{80 + 60}{2} = 70 \dots$$

また、これに出現エンジン数 $\times 0.1$ を加えることで、同じ得点の検索結果のうち、より多くの検索エンジンに出現した URL が上位に表示されるように総合得点を定義する。先述の例の場合、出現検索エンジン数は 4 であるから総合得点は 式のようになる。

$$point = 70 + 4 \times 0.1 = 70.4 \dots$$

4. クラスタリング

クラスタリング処理は、元のページを参照することなく、検索エンジンから取得したタイトル、サマリ、URL を評価することにより行う。その際、検索結果各々に、特徴量に相当するものとして、特徴語を設定する。特徴語は、そのページの内容を代表する語であると共に、クラスタラベルの候補となるものである。検索結果のクラスタリングをすることにより、膨大な検索結果を絞り込む際の支援、不要な検索結果の分類が可能になると考えられる。以下で、特徴語の設定方法と、クラスタの階層構造の生成方法を述べる。

4.1. 特徴語

以下のような、4 種類の特徴語を設けた。

- ・ 第一水準
経験則を利用

- ・ 第二水準
形態素解析の際に得られる品詞を利用
- ・ 第三水準
検索キーワードの前後に出現した名詞/複
合名詞を利用
- ・ 第四水準
検索結果のタイトル、サマリで最も先頭に
近い名詞/複合名詞を利用

次に各水準の特徴語の設定方法を述べる。

4.1.1. 第一水準特徴語 (経験則)

Web ページの中には、他の部類のページにはあまり出現しない特定のフレーズが含まれているなど、経験則によって Web ページの特徴を判断することが可能なページがいくつか存在する。そこで、経験則に基づいて特徴語を設定する。表 2 にその例を示す。

表 2 第一水準特徴語の例

特徴語	経験則
チャット	URL に 『chat』 を含む
掲示板	タイトル/サマリに 『投稿』/ 『re』 を含む URL に 『bbs』/ 『board』 を含む
シラバス	タイトル/サマリ/URL に 『シラバス』/ 『syllabus』 を含む
書籍	タイトル/サマリに 『isbn』/ 『頁』 を含む URL に isbn/book を含む
地域公式 サイト	URL に 『pref』 を含む
学術機関	URL に 『ac.jp』 を含む

4.1.2. 第二水準特徴語 (品詞情報)

茶筌 [14] を利用して形態素解析を行うと、組織名詞、地域名詞、人名名詞など品詞が細かく分類されるものがある。例えば、『成田宏和』を茶筌で形態素解析すると『成田』は姓名詞、『宏和』は名名詞として認識される。ここで、姓名詞、名名詞の順で連続して現れた場合は、一人の人物の名前であるとして、『成田』『宏和』を結合し、『成田宏和』という特徴語と、これが人物であるということで、『人物区分』という特徴語を設定する。

4.1.3. 第三水準特徴語 (検索キーワードの前後の名詞/複合名詞)

例えば、『東京』+ 『都立』+ 『大学』という検索キーワードでの AND 検索を考える。『東京』『都立』『大学』と絞り込み検索をしているとして『大学』を基にタイトル及びサマリを

評価し特徴語を設定する。この場合の特徴語は、『大学』前後の名詞/複合名詞で『大学』との間に含まれる形態素数が最も少ないものとし、等距離の場合は両方を特徴語とする。

4.1.4. 第四水準特徴語(最初に出現した名詞/複合名詞)

タイトルやサマリの先頭部分には、その検索結果の内容を反映する言葉が含まれている場合が多い。また、タイトルやサマリに検索キーワードが含まれていないなどの理由で、第三水準までの特徴語を設定できない場合がある。これらに対応するために、先頭から評価していき、最初に出来てきた名詞/複合名詞を特徴語として設定する。

4.2. 特徴語の設定例

『学生』というキーワードで検索した図4の検索結果について特徴語の設定例を考える。

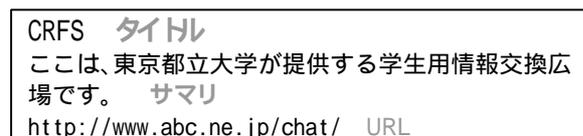


図4 検索結果例

まず、タイトルの『CRFS』を評価する。茶笥によるとサ変接続名詞として認識されるので、これを第四水準特徴語として設定する。続いて、サマリを評価する。『東京都立大学』が組織名詞として認識されるので、『組織区分』『東京都立大学』という二つの特徴語を第二水準特徴語として設定する。また、検索キーワード前後を見ると、前方のより後方が次に出てくる名詞との間に含まれる形態素数が少ない。『用』一般名詞、『情報』一般名詞、『交換』サ変接続名詞、『広場』一般名詞と連続した名詞が現れるので、これを複合名詞として『用情報交換広場』を第三水準特徴語として設定する。最後に、URLを評価する。URL中に『chat』というフレーズが含まれているので、『チャット』という第一水準特徴語を設定する。

4.3. 特徴語の得点付け

各水準の特徴語に重みを設定し、特徴語が設定されるたびに、その特徴語の得点に加えていく。全ての検索結果について、特徴語の設定が終わったら、各検索結果それぞれに設定されている特徴語のうち、もっとも得点の高いものを

代表特徴語とし、その検索結果の内容を代表している語であると判断する。

4.4. 木構造の生成

特徴語に基づいて、クラスタの階層構造生成を次に示す ~ の手順で行う。

各検索結果の代表特徴語のラベルが付いたクラスタに振り分ける。

要素が一つしか存在しないクラスタは『その他』のクラスタに振り分ける。

クラスタ名の前方一致により、クラスタ間の親子関係を決定する。

例えば、『メタ』と『メタ検索』というクラスタが存在する場合、『メタ検索』の親を『メタ』とする。

5. ユーザインタフェース

図5は、本システムをInternet Explorerで利用したときの表示画面、図6はNetscape



図5 ユーザインタフェース(IE)

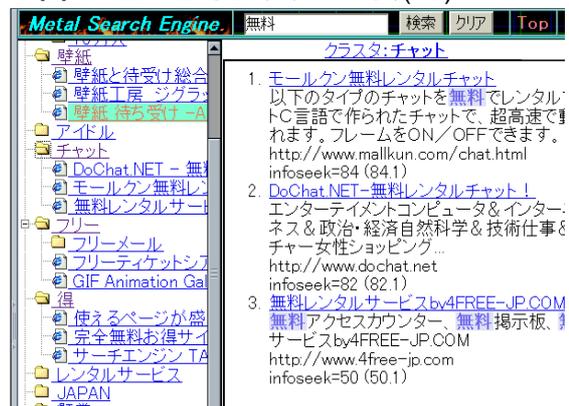


図6 ユーザインタフェース(NN)

- 📁, 📁 : フォルダのアイコン
- 🌐, 🌐 : ウェブページを示すアイコン

図7 アイコンの説明

Navigator で利用したときの表示画面を示す。上部のテキストボックスに検索キーワードを入力し、検索ボタン/Enter キーを押すと検索が開始される。スペースでキーワードを区切るにより AND 検索を行うことができる。左側のフレームには、階層的にクラスタリングされた結果を、Internet Explorer では HTML Help の機能、Netscape Navigator では Morten's JavaScript Tree Menu[15]を利用して表示する。フォルダのアイコンをクリックするとクラスタ内の検索結果が、Web ページを示すアイコンをクリックするとそのページが右側のフレームに表示される。

6. 実験と評価

6.1. 評価値の定義

まず、評価値を定義する前に、用いる用語の定義を表 3 の通り行う。

表 3 用語の定義

被評価検索エンジン	評価される検索エンジン
判定用検索エンジン	判定基準に用いる検索エンジン METALのクラスタリング機能を除いたもので、goo, Google, InfoSeek Japan, tocc で、メタ検索を行う
org	被評価検索エンジン上で『検索キーワード』で検索したときの検索結果 (検索結果全体)
clst	orgのうち、『クラスタラベル』のクラスタに含まれている検索結果 (或るクラスタに含まれている検索結果)
	orgのうち、判定用検索エンジンで『検索キーワード』+『クラスタラベル』でAND検索を行ったとき、その検索結果に含まれている検索結果 (そのクラスタに含まれるべき被評価検索エンジンの検索結果)
	のうち、clstに含まれている検索結果即ち、とclstの積集合(そのクラスタに含まれるべき検索結果のうち、実際そのクラスタに含まれている検索結果)
etc	『クラスタラベル』を『その他』としたときのclst

各記号を||で括るとその要素数を示す。

検索結果全体に於いて、或るクラスタに含まれるべき検索結果のうち、実際に含まれている検索結果の割合を示す指標として、**クラスタ再現率**を式により定義する。

$$recall = \frac{|b|}{|a|} \dots$$

或るクラスタに含まれている検索結果のうち、そのクラスタに含まれるべき検索結果の割合を示す指標として、**クラスタ適合率**を式により定義する。

$$precision = \frac{|b|}{|clst|} \dots$$

クラスタリングされた検索結果について、そのクラスタリングの精度を示す指標として**クラスタ値**を式により定義する。

$$100 \times \frac{\sum_{Cluster} (|clst| \times recall \times precision)}{\sum_{Cluster} |clst|} \dots$$

検索結果のうち、クラスタに振り分けられた検索結果の割合を示す指標として**クラスタリング率**を式により定義する

$$\frac{|org| - |etc|}{|org|} \dots$$

6.2. 実験と評価

検索エンジンのキーワードランキングで、よく上位に現れるとして『無料』『壁紙』『アイドル』、話題性で上位に出てきたキーワードとして『ワールドカップ』『チケット』という合計5つの検索キーワードについて、Lycos Japan, ナレッジエリサーチ, METAL にて検索を行った。この、クラスタリングされた検索結果の、第一階層のクラスタに対しクラスタ再現率、クラスタ適合率、クラスタ値、クラスタリング率を調査した。これらを表 5~7 に、5つの検索キーワードについての集計を表 8 に示す。実験の際の各システムの条件は、表 4 の通りである。

表 4 実験条件

システム名	検索に利用する検索エンジン	最大取得件数
ナレッジエリサーチ (KQS)	goo Yahoo! Japan(Powerd by Google) Excite(Powerd by Google) Infoseek Japan フレッシュアイ Lycos Japan	各25
METAL	goo Google Infoseek Japan tocc	各50
Lycos	Powerd by WiseNut	-

表 5 『無料』での検索結果(Lycos)

クラスタラベル	クラスタ内件数	クラスタ再現率	クラスタ適合率
Real com RealPlayer	2	100.0%	50.0%
完全無料	3	30.0%	100.0%
リンク集	7	42.3%	71.4%
無料WEB	4	15.0%	75.0%
無料レンタル	7	26.1%	85.7%
インターネットお	3	50.0%	33.3%
お小遣い	5	66.7%	80.0%
その他	92		
合計	123		

表 6 『無料』での検索結果(KQS)

クラスタラベル	クラスタ内件数	クラスタ再現率	クラスタ適合率
無料	13	16.9%	76.9%
チャット	9	42.1%	88.9%
無料レンタル 掲示板	2	11.8%	100.0%
簡単	3	28.6%	66.7%
無料 ホームページ	3	13.6%	100.0%
壁紙	2	40.0%	100.0%
ジャンル別 マガジン一覧	3	-	-
完全無料	2	33.3%	100.0%
Yahoo! Japan	2	0.0%	0.0%
その他	44		
合計	83		

6.3. 考察

表 5~7 は各システムで『無料』という検索キーワードで検索した場合の評価値である。表 8 は、5 つの検索キーワードについて、表 5~7 と同様に評価値を調査した際の集計結果である。再現率平均・適合率平均は、各システムの第一階層のクラスタの、クラスタ再現率・クラスタ適合率の平均を 5 つの検索キーワードで検索した場合それぞれについて更に平均を取った値である。クラスタ値・クラスタリング率は、5 つの検索キーワードで検索した場合それぞれについての、クラスタ値・クラスタリング率の平均である。

表 8 によれば、Lycos は、再現率に関してナレッジエリサーチの 27.9%、METAL の 28.7% に対し大きく差を付け 43.1% という値を示した。クラスタ率についても他と比べて良好な数値を出している。しかし、クラスタ率が最も低

表 7 『無料』での検索結果(METAL)

クラスタラベル	クラスタ内件数	クラスタ再現率	クラスタ適合率
掲示板	19	56.0%	73.7%
レンタル	9	18.8%	66.7%
Download	6	66.7%	66.7%
提供	5	10.0%	40.0%
得	4	40.0%	100.0%
メール	4	22.2%	100.0%
アクセスカウンタ	4	26.7%	100.0%
魔法	3	100.0%	100.0%
カウンタ	3	9.1%	66.7%
WEB	3	9.1%	66.7%
アイドル	3	30.0%	100.0%
プロバイダ	3	22.2%	66.7%
サービス	2	5.3%	100.0%
i-mode	2	0.0%	0.0%
HP	2	8.3%	100.0%
壁紙	2	25.0%	100.0%
翻訳	2	33.3%	100.0%
チャット	2	8.3%	100.0%
フリー	2	6.5%	100.0%
その他	42		
合計	122		

表 8 5 つの検索キーワードについての集計

システム名	再現率平均	適合率平均	クラスタ値	クラスタリング率
Lycos	43.1%	72.4%	34.5	42.0%
ナレッジ エリサーチ	27.9%	80.4%	18.8	54.0%
METAL	28.7%	83.3%	25.4	68.5%

く第一階層で『その他』以外のクラスタに振り分けられたのは全検索結果の 42% である。

ナレッジエリサーチは、『無料』という検索キーワードに対し『無料』というクラスタが出来てしまい、これは有効なクラスタとは言えない。他の検索キーワードについても同様、『検索キーワード』のラベルのクラスタが生成された。また、同じページを示す検索結果が同じクラスタ内に複数存在し、検索結果の統合がうまくいってなかった。再現率は最も低い値となったが、適合率は良好であった。

METAL は、クラスタリング率が最も高く Lycos、ナレッジエリサーチに比べ、より多くの検索結果が『その他』以外の第一階層のクラスタに振り分けられたことを示している。また、適合率も最も高く、不適切な検索結果がそのクラスタに振り分けられた数が少ないことを示している。

7. まとめ

本稿では、検索能力を強化すると共に、煩雑な検索結果から目的の情報を発見しやすくするために、検索結果をクラスタリングして表示するクラスタリング型メタ検索エンジン METAL を提案した。

本稿の実験により、Lycos, ナレッジクエリサーチに比べ、より多くの検索結果を『その他』以外の第一階層クラスタに振り分けることが出来、振り分けた際のノイズも少なくすることが出来た。

今後の課題として、『再現率』を高めることを考えている。検索エンジンで、『検索キーワード』+『クラスタラベル 1』と『検索キーワード』+『クラスタラベル 2』での検索結果が排他的なものではないので、今回のシステムのように、一つの検索結果が一つのクラスタにだけに属するようにしていると、クラスタ数が増えるほど再現率が低くなるものと思われる。そこで、一つの検索結果が複数のクラスタに属することが可能になるような手法を検討していく。

謝辞

本研究の一部は文部科学省科学研究費特定研究領域(2)「情報学:A02」(課題番号:14019075)による。

8. 参考文献

- [1]AltaVista
<http://altavista.com/>
- [2]HotBot
<http://hotbot.lycos.com/>
- [3]goo
<http://goo.ne.jp>
- [4]Google
<http://google.com>
- [5]Japan.internet.com
<http://japan.internet.com>
- [6]Vivisimo
<http://vivisimo.com>
<http://vivisimo.com/html/faq.html>
- [7]FUJITSU BUSINESS SYSTEMS LTD.
Knowledge Query Server 2.0
<http://kms.iws.ne.jp/qserver/default.html>
ナレッジクエリサーチ
<http://kd.iws.ne.jp/kms/kqs/q-search.html>

- [8]WiseNut
Japanese Engine
<http://www.wisenut.com/corp/jse.html>
New Search Engine WiseNut to Power Lycos Japan
<http://www.wisenut.com/corp/pdf/LycosWisenutPressRelease.pdf>
- [9]Lycos Japan
<http://lycos.co.jp>
- [10]Metcha Search
<http://bach.cs.kobe-u.ac.jp/metcha>
- [11]検索デスク
<http://www.searchdesk.com>
- [12]Infoseek
<http://infoseek.co.jp>
- [13]tooc
<http://www.tooc.co.jp/search/>
- [14]茶筌：奈良先端大学
<http://chasen.aist-nara.ac.jp>
- [15]Morten's JavaScript Tree Menu
<http://www.treemenu.com/>
- [16]安形 輝,石田 栄美,久野 高志,野末 道子,上田 修一：WWW ページの自動分類 NDC の分類体系と Yahoo のカテゴリを使った分類,情報処理学会研究報告(99-FI-54) . Vol.99, No.39, p.113-120(1999-05-17)
http://www.slis.keio.ac.jp/~ueda/ueda_www2.pdf
- [17]佐々木 稔,北 研二：文脈関連度による検索質問の関連語抽出,言語処理学会第7回年次大会, pp.105-108, March (2001).
http://www-a2k.is.tokushima-u.ac.jp/member/sasaki/frame_home/Papers/NLP01.ps
- [18]鈴木 優,田中 克己,吉川 正俊,植村 俊亮
日本ソフトウェア科学会 第2回インターネットテクノロジーワークショップ(WIT'99), pp.156-162, 1999年8月.
<http://www.brl.ntt.co.jp/itech/wit99/proceedings/suzuki/suzuki.ps.gz>