ブックマーク連携型検索エンジン「netPlant」

大澤 昇平
ohsawide@simfan.cn1.jp
福島工業高等専門学校

現在普及している検索エンジンの問題点は、「良いキーワードが思いつかないと、良い検索結果が返ってこない」ことである。そこで本稿では、「同じ目的を持った人たちであれば、最終的に辿り着くページもほとんど一緒である」という仮定に基づき、上記のような問題点の無い検索エンジン、「netPlant」の提案を行う。

Bookmark-Associated Search Engine “netPlant”

SHOHEI OHSAWA
Fukushima National Collage of Technology

1 序論 —キーワード型検索エンジンの功罪—

インターネットを利用した情報収集が大衆化してきた昨今、検索エンジンにはより個人の目的に合った情報を返すことが要求されてきている。ところが、いくら検索エンジン側の精度を高めても、良いキーワードを考えられるユーザと考えられないユーザとでは、目的のページに辿りつくまでの時間にどうしても差が出てきてしまう。つまり、検索エンジンの実効的な精度が、ユーザの検索技術に依存しているのが現状である。これは、近年の「キーワード型検索エンジン」の普及に起因していると考えられる。

かつて「ディレクトリ型検索エンジン」が主流であった頃には、ページが人間の手によって階層的に分類されていたため、たとえば「クラシック音楽について調べたい」という漠然とした要求に対しても、検索エンジンは質の良いページを返すことが可能であった。しかし、ディレクトリ検索エンジンは今なお拡大し続けるウェブページの数に対応できなくなり、最近ではキーワード型にその座を奪われることとなった。

ところがキーワード型は、より多くのページを網羅しているものの、玉石混交としたウェブページの中からユーザのニーズに合ったページを抽出するための機構がディレクトリ型ほど上質ではないため、ユーザは目的のページに辿り着くために、クエリを工夫する必要性を迫られるようになった。そして、クエリを工夫できるユーザとできないユーザとで、目的のページに辿り着けるか否かが別れるようになり、現在に至るのである。

そこで本稿では、「同じ目的を持った人間であれば、最終的に辿りつくページも同じ」という仮定に基づき、現在普及している検索エンジンの問題点を補うべく、ウェブの拡大に追従できるディレクトリ型検索エンジン「netPlant」を提案する。

2 従来のディレクトリ型検索エンジンの課題

Yahoo!などの従来のディレクトリ型検索エンジンは、人力でページの分類を行っているため、「返される検索結果の質が高い」という利点があったが、一方で「検索対象になるウェブページ数が限られる」「あらゆるウェブページを即時に検索結果に反映できない」などの問題点が

あった。また、運営時にはウェブページの分類を行う専用の人員（「サーファ」）を雇用する必要があるため、人件費など運営上のコストが高いという問題点もあった。

ディレクトリ型検索エンジンが、昨今のウェブの拡大に伴って衰退するようになったのは、このような問題点が対象となるウェブページ数の増加に比例して深刻化する傾向があるからであると考えられる。

したがって、現在においてディレクトリ型検索エンジンを運営する場合、まずウェブページ数の増加に比例して深刻化する問題を解決する必要がある。この問題だけを解決するには、ページの分類を人工知能的に行えば良いわけであるが、これではディレクトリ型検索特有の質の高さを失われてしまうと考えられる。

以上のことを踏まえると、新型ディレクトリ型検索エンジンに要求される必要条件は次のようになる。

1) ページが階層的に分類されている
2) 人力による分類である
3) 質の高いページが上位に表示される
4) 対象となるウェブページ数の増加に追従できる

3 「マクロブックマーク」

従来のディレクトリ型検索エンジンの問題点は、ページの分類を行う人間が特定の集団に限られることに起因していると考えられる。したがって、ページの分類を行う人間を特定の集団に限らないようにすれば、問題は解決できると考えられる。

そこで本節では、Web2.0 的な手法を用いることで、ページの分類を検索エンジンを使うユーザ自身に行わせる仕組み「マクロブックマーク」について提案する。

3.1 概要

マクロブックマークとは、不特定多数のユーザのブックマークを統計処理することで、事実上一つの巨大なブックマークとして存在するかのように振舞う仮想的なブックマークである。

統計処理は単純な重ねあわせによるもので、たとえばユーザ A、B、C がそれぞれ図 3.1 のようなブックマークを持っていた場合、生成されるマクロブックマークは図 3.2 のようになる。

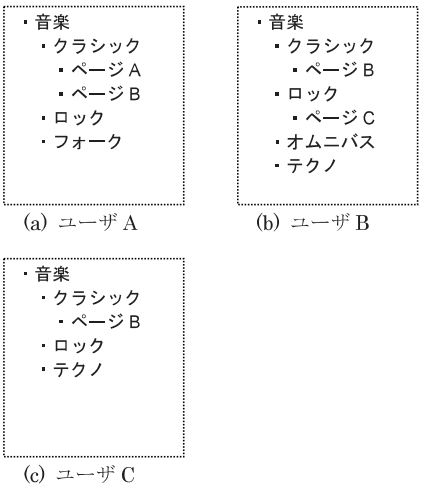


図 3.1　各ユーザのブックマーク

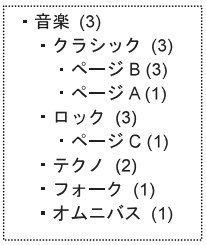
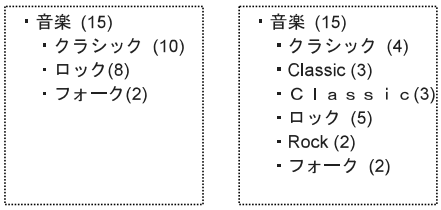


図 3.2　生成されるマクロブックマーク

ここで、括弧内の数字は重複の個数を表している。重複の多いカテゴリやページ（以下、総称して「ノード」）は人気が高いと考えられるので、これを「推奨度」というパラメータとして考え、閲覧時には上位に表示するようにする。

ただし、単にブックマークを重ね合わせただけでは、集計結果に相当なバラツキが生じる可能性が高い。図 3.3

に例を示す。



(a) 少ない　(b) 多い

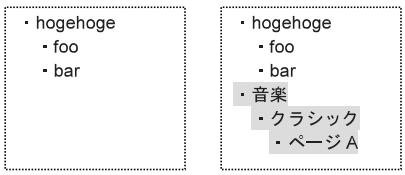
図３．３　バラツキの量の比較

(a)の場合は、表現のゆれが少ないのに対して、(b)の場合は表現のゆれが多い。後者の場合、同じクラシックに関するブックマークでも“クラシック”“Classic”“Ｃｌａｓｓｉｃ”など表現が多様化してしまうため、本質的な推奨度が反映されていない。

そこでマクロブックマークでは、次のような工夫を行うことで、表現のゆれが少なくなるようにしている。

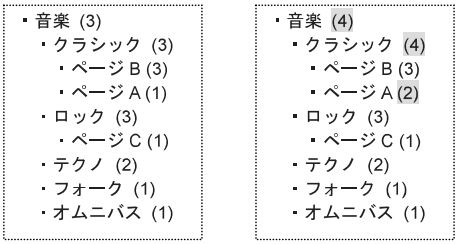
3.2 閲覧者を利用したフィードバック

マクロブックマークでは、閲覧者が開いたノードが、自動的に本人のブックマークに登録されるようになっている。これにより、閲覧者は結果的にそのノードへの投票を行うことになる。なぜならノードがブックマークされることによって、そのノードのマクロブックマークにおける重複（i.e.推奨度）が増加するからである。たとえばユーザ D が図 3.2 のマクロブックマークのうち“音楽>クラシック>ページ A”を開いた場合、ユーザ D のブックマークは図 3.4 のように変化する。したがって、マクロブックマークは図 3.5 のように変化する。



(a) 変化前　(b) 変化後

図 3.4　ノード閲覧によるブックマークの変化



(a) 変化前　(b) 変化後

図 3.5　ノード閲覧によるマクロブックマークの変化

図のように、ユーザが閲覧した部分の推奨度が高くなるようになる。ユーザは既存のノードから選択する形で自分のブックマークへの登録を行うため、ここで表現のゆれは生じない。

また、誤って興味のないノードを開いた場合でも、ユーザは自分のブックマークからそのノードを削除することができる。これは、結果的にはそのノードへの投票を無効にすることに相当する。このようにして、最終的には人気のあるノード、即ちより多くのユーザにブックマークされたノードが上位に表示されることになる。

3.3 新型ディレクトリ型検索エンジンとしての利用

ここで、第 2 節の最後に述べた新型ディレクトリ検索エンジンの必要条件に、マクロブックマークを照らし合わせてみる。

1) ページが階層的に分類されている -- ブックマークを階層的に行わないユーザは少ない。

2) 人力による分類である -- これは自明である。

3) 質の高いページが上位に表示される-- 推奨度がページの質に相当すると考えられる。

4) 対象となるウェブページ数の増加に追従できる -- 不特定多数のユーザの力を利用しているため可能。

このように、マクロブックマークは新型ディレクトリ型検索エンジンの必要条件を満たすため、新型ディレク

トリ型検索エンジンとして利用できる可能性があると考えられる。

3.4 正のスパイラルモデル

マクロブックマークは、ユーザ個人に対してはディレクトリ検索エンジンとしての検索結果を提供し、逆にユーザ個人からはブックマークのデータを取得する。このような、利用者と運営者が互いにリソースを提供しあうというモデルを採用することによって、マクロブックマークは、ユーザが使えば使うほど成長するという、「正のスパイラルモデル」を呈することになる。

3.5 Yahoo!に対する優位性と弱点

このような仕組みにより、マクロブックマークはサーファの不要なディレクトリ型検索エンジンとして機能するだけでなく、検索結果にユーザ自身が関与することによって、カテゴリの種類やページの推奨度などが、よりユーザの好みに合ったものに最適化されていくと考えられる。これは、検索結果がサーファの趣向によってのみ決定されていたYahoo!に対する優位性である。

また、マクロブックマークはそれ自体が「共有ブックマーク」として機能するため、ユーザに一人で調べる以上に情報が集まってくるというメリットを与えることができる。これについては第5節で具体例を述べるので参考にされたい。

逆にマクロブックマークの弱点は、初期状態では検索に必要なデータが全く存在しないという点である。このような運営上の問題点に関しては、第4節で実装レベルでの対応策を述べることにする。

4 システム概要

以上のような理論を踏まえて、マクロブックマークを実装した検索エンジン「netPlant」について提案する。

4.1 マクロブックマーク機能

netPlant 上で“プログラミング”と検索すると、図4.1 のような検索結果が表示される。カテゴリ“プログラミング”に分類されるサブカテゴリが放射状に表示される他、右のペインには同カテゴリに分類されるページが表示される。このデータは、マクロブックマークから抽出されたものである。ページは推奨度順にソートされており、推奨度が高いものが上位に表示される。ユーザは表示されたカテゴリを自分の目的に応じて選択していくだけで、目的のページに辿り着ける仕組みになっている。

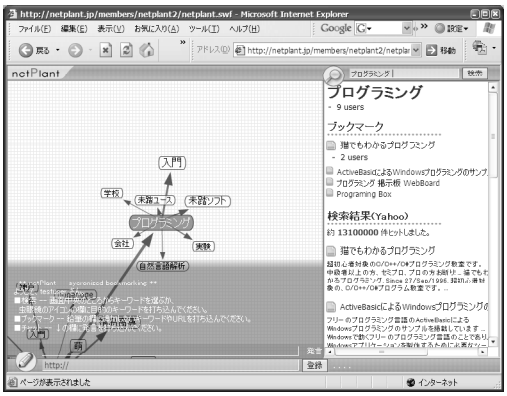


図 4.1 “プログラミング”での検索結果

また、以上のような検索結果はユーザ自身のブックマークという形でロギングされ、マクロブックマークへとフィードバックされる。もちろん、ユーザが独自のカテゴリやページを登録することも可能であり、この行為によって、マクロブックマーク内のデータが多様化することになる。

4.2 サーチアシスタント機能

前節でも述べたとおり、マクロブックマークには、最初の状態ではデータが全く存在しないという致命的な欠点が存在する。そこで netPlant では、この欠点を補う機能として、「サーチアシスタント機能」を搭載している。

サーチアシスタントとは、通常の検索エンジンで検索

する際に入力した検索クエリをカテゴリと見なしたマクロブックマークである。

たとえば、“プログラミング 入門 言語”と検索してページ http://www.hogehoge.com/を開いた場合、ユーザのブックマークにはカテゴリ“プログラミング > 入門 > 言語”が自動作成され、その中に当該ページが登録されることになる。するとその後に検索したユーザは“プログラミング”と入力するだけで、クエリ“プログラミング 入門”を netPlant に提案されることになる。

サーチアシスタントの便利な点は、複数ユーザで検索クエリの同期が取れる点である。同じ目的でページを探す場合でも、ユーザによって検索クエリの作り方は異なる。これによって、目的のページに辿り着けるユーザと辿り着けないユーザが出てきてしまう。しかし、以前に目的のページに辿り着くことに成功したユーザの検索クエリを利用すれば、通常の検索エンジンでクエリを上手く作れないユーザでも目的のページに辿り着くことが可能になると考えられる。

実際に運営を行う際は、まずサーチアシスタントでユーザの囲い込みを行い、徐々にマクロブックマークのデータベースを成長させるという手法をとることを予定している。

4.3 新着ブックマークの通知機能

netPlant には、ユーザが自分のブックマークに登録しているカテゴリに他のユーザがページを登録した場合、通知を行う機能が搭載されている。この機能を用いることによって、ブックマークの共有を行ったり、複数人同時にブックマーキングを行ったりすることが可能になると考えられる（第 5 節で詳述）。

5 ユースケース

本節では、想定しているユースケースを示す。

例 1 初学者による検索

「プログラミングの初学者が、自分に合った言語を探したい」というようなシチュエーションを考える。このような場合、ユーザは特に深く考えず、“プログラミング”と netPlant に入力するだけで良い。すると、netPlant は“プログラミング”というカテゴリの内部を表示する。ユーザはその中から、サブカテゴリ“入門”を選択する。同様にしてカテゴリ“プログラミング>入門>言語”を開いたユーザは、プログラミングの初学者でもわかる言語に、「BASIC」や「Java」や「なでしこ」があることを知る。ユーザはこの中からカテゴリを選択することで、最終的に目的のページ、「初心者向けのプログラミング言語に関するページ」に辿り着くことが可能となる（図 5.1）。

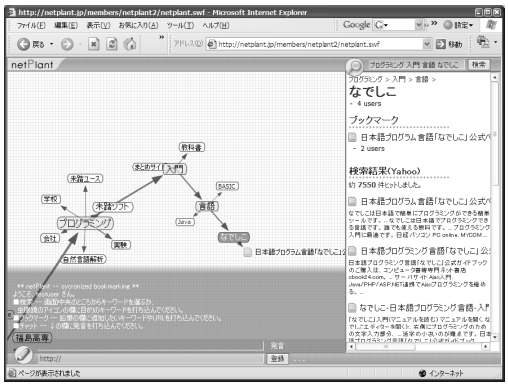


図 5.1 検索例

これと同じ作業を Google などのキーワード型検索エンジンで行おうとした場合、ありとあらゆるキーワードを試していかなければならない。しかし netPlant では、ユーザが考えるべきキーワードは“プログラミング”の一言で済むのである。

例 2 ブックマークの共有

まず、ユーザ A がとあるカテゴリやページをブックマークしておく。すると、マクロブックマーク上で同じノードを開いたユーザ B がそれを自分のブックマークに登録し、さらにそのカテゴリ内に独自のページを登録する

可能性がある。するとこの場合、後でユーザ A がブックマークを見返してみたときに、カテゴリ内のページが増えていることになる。つまり、一人で調べる以上に情報が集まるわけである。この現象をユーザ A の視点から観測した場合、放っておいたブックマークが、あたかも成長しているかのように感じられる（これが、本システムが「netPlant」という名称を持つ理由である）。

例 3　複数人同時ブックマーキング

netPlant には、ブックマークを同期する機能がある。たとえば、“プログラミング” というカテゴリをユーザ A と B がブックマークしていた場合、ユーザ A が同カテゴリ内にノードを追加した際に、ユーザ B のブックマークに即座に反映されることになる。

この機能を利用することで、例えばレポートを課せられた学生が、複数人で一緒に調べ物をするといったことが可能になると考えられる。

6 類似のシステム

netPlant にコンセプトが類似したシステムについて挙げる。

はてなブックマーク[1]

ソーシャル・ブックマークと呼ばれるブックマークサービスで、ユーザのブックマークを統計処理するというコンセプトが netPlant と類似している。しかし、ブックマークの共有とトレンドの発掘を重視した設計であるため、検索エンジンとしての実用性は低い。

vivisimo[2]、あ～るNavi[3]

人工知能によってページのカテゴライズを行うタイプの検索エンジン。ページのカテゴライズを行うという点で netPlant と類似しているが、アプローチが人工知能的であるのが特徴。

S-team.jp/Search[4]

ユーザがページを登録していくタイプのディレクトリ型検索エンジン。ページの登録をユーザに委ねている点では netPlant と共通しているが、ページを篩いにかける機構を設けていないため、ページの質は低い。

Googleサジェスト[5]、Ask.jp[6]

検索エンジン側がユーザにクエリを提案するタイプの検索エンジン。netPlant のサーチアシスタント機能と類似。

7 おわりに

netPlant は、検索を行うユーザ自身の力を利用して成長する、Web2.0 的な検索エンジンである。

今後の課題としては、ツールとしての実用性の向上を目的として、ユーザ・インターフェースを重点的に改良していきたい。

参考文献

[1] http://b.hatena.ne.jp/
[2] http://vivisimo.com/
[3] 小澤崇記：ブラウジングしているページを自動連想検索するインタフェース(2005)、筑波大学第三学群情報学類卒業研究論文
[4] http://www.s-team.jp/search/
[5] http://www.google.co.jp/webhp?complete=1&hl=ja
[6] http://ask.jp/