

セマンティック Web に向けての Web ブラウザによる意味的処理のイメージ

平松宙祥, 三浦浩一, 松田憲幸, 瀧寛和, (和歌山大学大学院システム工学研究科)
安部憲広(九州工業大学情報工学部), 堀聡(ものづくり大学)

連絡先：〒640-8510 和歌山県和歌山市栄谷 930
e-mail：s041051@sys.wakayama-u.ac.jp, taki-lab@sys.wakayama-u.ac.jp

概要

セマンティック Web のメタデータを参照して検索を行うブラウザを開発した。メタデータの意味情報を利用して、意味検索を実現する。通常の Web とセマンティック Web を連携して検索する方式を提案する。

1. はじめに

現在、WWW(以下Webと呼ぶ)には、多種多様な情報が存在しており、情報が多すぎるために必要とする情報を探し出すことが困難になっている。その解決策として現在注目を浴びているのがセマンティックWebという技術である。セマンティックWebはWeb上の情報に対して機械的に処理可能にするための意味づけをする技術であり、実用化に向けて様々な研究がなされている。しかし、それらの研究のほとんどは、いかにして意味情報を記述するかを対象にしている。たとえば、意味を定義するメタデータの入力を補助する方法であるとか、オントロジーの構築方法であるとか、それらの記述言語についての研究が大半を占める。本研究では、それらの部分は余り気にせず、実際にそのようにして記述さ

れたメタデータがすでに存在するという前提で、そのメタデータの運用に関する研究を行う。メタデータを用いる利点の一つには、その意味の情報を利用することによって、検索の効率や正当性が向上すると言われている。そこで、メタデータを用いた意味処理として検索システムを考える。具体的には、個人または企業が作成した「書籍のレビュー」とそれに関するDublin Coreスキーマ[1]で記述されたメタデータを、ユーザがブラウジングした履歴と、リンク構造や専用のタグで記述された情報を元に先読みしたページから収集し、クライアント側での検索におけるメタデータの運用方法を提案する。この手法はRDFによるサイトサマリー(RSS[2])等の他の形式のメタデータにも応用することができる。

2. 検索対象メタデータの収集

検索を行うには、検索対象となるメタデータを収集する必要がある。本章ではその方法について述べる。

まず、前提として、ウェブにコンテンツを公開しているオーサーは、そのページに記載されている情報に関するメタデータを何らかの形(ここでは RDF とする)で記述し、ネットワークからアクセス可能な状態にしてあるものとする。RDF ファイルは関連づけられているファイルのファイル名の末尾に ".rdf" をつけて HTTP のサーバー上に置かれているものとする。

現在、ウェブをブラウジングする場合には検索エンジンやポータルサイトを利用する。本研究のシステムも使う場合も同じで、情報検索者(ユーザ)は、例えば「書籍レビュー」と入力してウェブを検索し、書籍のレビューサイトに到達する。しばらくは普通にブラウジングしていたが、ユーザは昨日買った本のことを思いだし、それについてのレビューを調べたいと思った。ユーザは今レビューサイトを見ているので、トップページへ戻りリンクをたどるか、目次ページを検索することによって目的とするページに行く。しかし、リンクをたどるのが煩わしかったり、現状のブラウザによる文字列検索では、単純なマッチングのみを行うためにしばしば全く関係のないところを見ることになる。また、一回の検索で済まず、いくつかのページに渡って検索を行ったり、検索サイトに戻ったりする必要が出てくるかもしれない。そこで、システムでは、「現在見ているページ」と「今まで見てきたページ」、「これから見られるページ」の三つを検索対象にできるように

なっている。つまり、ブラウジング履歴とブックマーク、リンクの先読みによって HTML のページと RDF によるメタデータを収集し、検索対象とする。

3. 検索対象となるファイルの選択と検索キーワードの入力

ユーザが検索開始ボタンをクリックすると、検索範囲選択ウインドウ(Fig.1)が現れる。

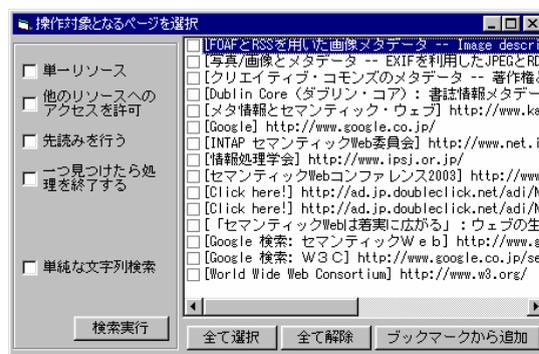


Fig.1 検索範囲選択ウインドウ

履歴は多すぎて困るということはないので基本的には全て使用するが、ブラウジングの途中で目的が変わることは頻繁にあり得るために特定のページを対象から外すことが可能になっている。

「ブックマークから追加」のボタンをクリックすればブックマークからページを追加することができる。主にブラウジング開始時に使用することを想定している。

「先読みを行う」のチェックを入れると、現在表示しているページのリンクをたどってそのページのメタデータも検索対象とする。

「一つ見つけたら処理を終了」のチェッ

クを入れると、検索によってヒットした始めの一つを表示する。そこに表示される「次」のリンクをクリックすることで一つずつ表示していく方式になる。

「検索実行」のボタンをクリックすると、検索キーワード入力ウインドウ(Fig.2)が現れる



Fig.2 検索キーワード入力ウインドウ

知識（意味情報）を利用した検索を行うために、メタデータの項目ごと（例えば、Dublin Core のスキーマで書かれたメタデータの場合はタイトルや著者名）ごとにキーワードを設定する。Fig.2 のような入力フォームは、DTD から自動的に抽出される。ある項目が文字列なら文字列のマッチングになるし、数字である場合は、等号や不等号を用いて“範囲”を記述することができる。また、性別のように特定の値以外取らない項目は選択式のリストとして表示され、そこから選択することになる(Fig.3)。



Fig.3 選択式リストの例

4 . 検索結果の表示

検索結果の表示には2種類の方法を用意する。一つは、3章で指定したメタデータを全て同時に検索し、その結果を表にして表示する方法である。もう一つの方法は、最近参照したページのメタデータから順に検索を行い、一つ見つかったらそれを表示して、ユーザーの操作によって次の検索を行う方法である。前者はサーチエンジンを利用した検索による検索結果と同じような形になり、後者はブラウザによる文字列検索に相当する。

前者の方法では、Dublin Core のメタデータの場合、基本的な要素タイプの場合でも、15列の表になってしまう。ほとんどの環境では、このような表を一画面に表示することができず一列の幅が非常に小さくなってしまいか、横にスクロールさせる必要があり、視認性に欠けるため、表示する列を選択式にするなどのユーザーインターフェースが必要となる。

どちらの場合も、検索結果からは、そのメタデータがつけられているURIへのリンクが表示され、そのリンクをクリックすることによって目的とするページへ移動する。

	title	creator	subject	description	publis
09	知識の表現と利用	上野青樹			オー
10	知識の獲得と学習	大須賀節雄			オー

Fig.4 検索結果の表による表示例

5 . ブラウザによって処理を行う意味

本研究の前提となっている、すでにメタデータが存在する環境においては、検索工

ンジンからもその情報を利用可能なため、ブラウザによってこのような処理を行う必要がないように思われるかもしれない。しかし、現実には、オーサーが記述した内容が検索エンジンに反映されるまでにはある程度の時間が必要であるため、「ここに行けばこれに関する新しい情報が載っているだろう」とユーザが知っている情報、あるいは「お気に入りのサイト」の情報ならば、検索エンジンによる検索よりも早い段階で同じ結果を得ることができるため、このようなシステムは有効である。

6 . 今後の予定

上記のような機能の他に、次のような機能を考えている。

6 . 1 履歴の自動選択

現在は URL 一つ一つに対して使用するかどうかのチェックを入れなければならないが、この作業は非常に手間がかかり、場合によっては検索結果が望むものにならない場合がある。そこで、現在表示しているページに関連づけられているメタデータと同様の形式、または互換性のある形式のメタデータを持っているページのみを表示する機能を考えている。

6 . 2 汎用性の向上

今回のシステムは「書籍のレビューサイトを検索する」というシナリオを想定して実装を進めたため、メタデータの形式として Dublin Core を用いた。そのため、メタデータが階層状や複雑なネットワーク状になっていることを想定していない。メタデ

ータがそのように記述されている場合についての取り扱いについては、現在思案中である。

7 . まとめ

ユーザのブラウジングの履歴とブックマーク、リンクの先読みによってウェブページに関連づけられているメタデータを収集し、検索処理を行う手法を提案した。この手法を用いれば、通常の全文検索では不可能な、意味情報を用いた検索をブラウザによって行うことができる。検索サーバーでも同様の処理を行うことはできるが、クライアント(ブラウザ)で行うことによって、検索サーバーが検索対象ページを取得するまでのタイムラグをなくすことができるため、このシステムは有用であるといえる。

8 . 参考文献

[1]DCMI, Dublin Core Metadata Element Set, Version 1.1: Reference Description, 1999, DCMI Recommendation

<http://dublincore.org/documents/dces/>

[2]Dan Brickley, et al., RDF Site Summary (RSS) 1.0, 2000, RSS-DEV Working Group

<http://purl.org/rss/1.0/spec>