インターネット検索の原理とメタデータの利用

兼宗 進 (kanemune@cc.hit-u.ac.jp)
一橋大学 総合情報処理センター

概 要

インターネットにおける情報検索の流れは、過去に人手によるディレクトリインデックスの構築から検索エンジンによる全文検索にシフトしてきた経緯がある。本発表の前半では、主に検索エンジンの索引構築と検索の仕組みを概観し、その工夫と構造上の限界を明らかにする。後半では、従来の検索サービスでは扱うことが難しかった、更新が頻繁に行われるページの例としてニュースサイト、Wiki、Blog などを紹介し、サイトの更新情報を公開する RSS の仕組みを紹介する。RSS はサイトの要約を扱うメタデータであり、Dublin Core を包含することができる。この RSS を図書館に応用した例として、一橋大学附属図書館のサイト更新情報公開サービスと、今回試作した新着資料公開システムを紹介する。

1 検索エンジンの仕組みとその限界

1.1 インターネット検索の歴史

インターネットにおける情報検索の流れは、過去に人手によるディレクトリインデックスの構築から検索エンジンによる全文検索にシフトしてきた [12]。表 1 に、ディレクトリインデックスと検索エンジンの特徴を示す。

表 1: ディレクトリインデックスと検索エンジン

	利点	限界
ディレクトリインデックス	人手による厳選 階層構造	数が少ない 登録・更新が遅い (月単位) サイト単位
検索エンジン	ページ単位 ランキング 更新が早い (日単位)	

1.2 検索エンジンの仕組み

検索エンジンの本質は、「先頭ページ (10 件程度) に有用なページを集められるか」にある。そのために、検索結果のランキング処理が命である (正確には先頭数ページのランキングであり、それ以降の順位は大勢に影響しない)。また、先頭ページに十分な数の有用な

表 2: 検索漏れの種類

検索漏れ	説明
収集	対象のページを収集して索引に格納できない
全文検索	索引から検索できない
ランキング	有用なページが先頭ページに来ない

ページを表示できるのであれば、多少の検索漏れは問題にならない。表 2 に検索漏れの例を示す。

検索エンジンは、収集処理で WWW を巡回し、あらかじめ多くの WWW ページを収集して索引に蓄えておく。ユーザーが検索を行うと、索引から検索語を含む検索処理が行われ、結果集合が作られる。結果集合にはランキング処理が行われ、重要度の高いページから結果が表示される。

表 3: 検索エンジンの索引生成処理

処理	内容
収集	WWW を巡回し、ページを収集する
テキスト整形	文字の表記を加工する
索引語生成	テキストを分割し、索引語を作る
索引構築	索引に索引語を格納する

表 3 に索引生成処理の概要を示す。収集処理では、ロボット (robot) と呼ばれるプログラムを実行し、WWW を巡回してページを収集する。ロボットは、自律的に WWW ページを閲覧して回ることから、スパイダー (spider) やクローラー (crawler) と呼ばれることもある。ロボットは過去に収集したページに対して、定期的に巡回を繰り返す。巡回の間隔は短いことが望ましいが、検索エンジンが収集対象とするページは現在数十億ページに達しており[1]、すべてのページを短期間で巡回することは不可能である。そこで、「ページの更新頻度に応じて巡回する」、「ページの重要度に応じて巡回する」などの巡回戦略が用いられている。

WWW ページを収集した後は、HTML(Hyper Text Markup Language) からテキストを取り出す。大量のデータから高速に検索を行うために、WWW ページから取り出したテキストは、分割して索引に登録される。代表的な索引語生成手法として、形態素解析と N-gram が存在する。

表 4: 検索エンジンの検索処理

処理	内容
全文検索	高速にテキスト検索を行う
ランキング	有用なページを先頭に集める
表示	ページ情報を数行に要約して示す

[1]Google の収集ページ数は、2004 年 7 月現在で約 43 億ページである。

表 4 に検索処理の概要を示す。ユーザーから検索語を受け取ると、検索エンジンは索引を用いて検索を行い、結果集合を作る。ランキングを行うために、検索結果に重み付けを行う処理をスコアリングと呼ぶ。

1.2.1 検索エンジンの限界

収集面から分類した Web ページの種類を表 5 に示す。

表 5: Web ページの種類

種類	検索エンジンの対応
更新頻度の低いページ (A)	○
更新頻度の高いページ (B)	△
DB から生成されるページ (C)	×
他からリンクされないページ (D)	×

検索エンジンの Web ページに対する巡回と索引の更新は、数日から数週間程度かかる。このような性質は、いちど作られるとしばらくそのまま公開される従来の更新頻度の低いページ (A) に対しては有効であったが、近年増えている時間単位で更新される更新頻度の高いページ (B) に対しては有効でない。また、データベースからリクエストに応じて生成されるページ (C) をあらかじめ収集して索引に登録することは困難であり、他からリンクされていないページ (D) を収集することは現在は不可能である。

特に 2000 年代になり各種のコンテンツ作成環境の発達により、Wiki や Blog といった更新頻度の高いページ (B) の比率が高まって来た。このようなページの検索への対応が不十分であることは、現在の検索エンジンの課題のひとつである。

2 新しい WWW の流れと RSS

後半では、従来の検索サービスでは扱うことが難しかった、更新が頻繁に行われるページの例としてニュースサイト、Wiki、Blog などを紹介し、サイトの更新情報を公開する RSS の仕組みを紹介する。RSS はサイトの要約を扱うメタデータであり、Dublin Core を包含することができる。

2.1 更新が頻繁に行われるページの例

1990 年代は static な Web ページ (更新のほとんどない Web ページ) が主流であったが、2000 年代になって頻繁に更新される Web ページが増えて来た。このようなサイトとしては、ニュースサイト、Blog、Wiki が代表的である。

ニュースサイト[2]は、新聞社やニュース配信企業を中心に、従来は新聞や TV、ラジオなどで伝えていたニュースをオンラインで提供するサービスである。

[2]RSS を配信するニュースサイトは Bulknews[2] の RSS Feed リストなどで見ることができる。

Blog は Weblog の略で、継続的に更新されるサイトである。個人で運営され、ニュースや日記、他のサイトへのコメントなどを記述することが多い[3]。

Wiki [4] は Web ブラウザからページを作成・更新できるシステムで、複数の人が共同で管理することを想定している。

これらのサイトは刻々と変化する性質を持っており、日単位で更新される検索エンジンの訪問を待っていることはできない。そこで、多くのサイトでは自前でサイトのサマリ (更新情報) を RSS の形式で公開している。

これらの仕組みに共通するのは、CMS(Contents Managemant System) を利用することで、機械的に Web ページ (HTML) を生成していることである。サイトを更新するたびに RSS を手作業で記述することは現実的ではない。RSS は Blog をはじめとする HTML 生成技術の発展により急速に普及した。

2.2 RSS

RSS[10] とは、サイトのコンテンツ情報を公開するためのデータ形式である。

現在、RSS には複数のバージョンが存在する。Really Simple Syndication, RDF Site Summary と RSS の略語の定義が異なっていることからわかるように、歴史的な事情から複数の規格が併存している状態である[5]。

RSS は XML で記述する。HTML が人間に対して情報を表示することを目的とするのに対し、XML はシステムに対して情報を伝えることを目的とするデータ形式である。

RSS ではサイト情報をチャネル (channel) で、個々のコンテンツ情報をアイテム (item) で表現する。

2.2.1 Really Simple Syndication

RSS0.91[3] は、世界中で使わるきっかけとなった RSS のバージョンである。歴史的に古く多くの実績があるが、メタデータの記述は貧弱である。item の数は 15 個以内に制限されており、item に書けるのは title、link、description だけである。このような理由から、多くの RSS リーダーは title、link、description だけを表示するものが多い。現在は RSS2.0[5] に発展している。図 1 に、RSS0.91 の記述例を示す。

2.2.2 RDF Site Summary

RSS1.00[4] は RDF(Resource Description Framework)[11] で記述された RSS のバージョンである。要素に DC(Dublin Core) を使えるなど、メタデータの記述は豊富である。日

[3]Blog は他のニュースサイトへの独自のコメントを掲載する形で発展した経緯がある。図書館関係では、司書の目と耳 [13]、図書館日記 [15] などが典型的である。

[4]Wiki はオリジナル [9] が提唱された当時からページを共同で管理することを特徴としている。TP&D フォーラムのサイト [7] は PukiWiki という Wiki のクローンで管理されているが、現在は管理者のみが修正できるようになっており、Wiki 的ではない。

[5]RSS は、0.9x から 2.0 に発展したが、別の規格として 1.0 が存在するなど、数字がバージョンの関係を表さないことに注意が必要である。最近ではこれらを統合する規格として ATOM[1] などが検討されている。RSS 全般については、The Web KANZAKI[14] に詳しい解説がある。

本で多く使われている。RDF はメタデータを XML で記述するための枠組みである。

RSS1.0 では、モジュール (module) の概念により、拡張を行える。RSS1.0 の item には title、link、description だけを記述できるが、mod_dublincore を宣言することで、Dublin Core の要素 (element) である dc:title, dc:creator, dc:subject, dc:description, dc:publisher, dc:contributor, dc:date, dc:type, dc:format, dc:identifier, dc:source, dc:language, dc:relation, dc:coverage, dc:right を記述できるようになる。図 2 に、RSS1.0 の記述例を示す。

2.3 RSS の利用

RSS は Web サーバー上で公開し、URL によってアクセスする。WWW サイトにアクセスするために Yahoo や Google が必要なように、RSS にアクセスするためにも、名前や分類などの情報から URL に変換するための仕組みが必要である。

RSS を公開するサイトでは、次の形で RSS の存在と位置を示すことが多い。

- サイトの HTML には、META 情報として RSS の URL を記述しておく。
- HTML 中に、RSS へのリンクを埋めたアイコンを表示しておく。

RSS をクライアントからアクセスするには、RSS リーダーを使うことが多い。複数の RSS に定期的にアクセスし、更新情報を取得する。RSS リーダーを使うことで、更新されたサイトの最新情報を取得し、コンテンツのメタデータ (タイトルなど) を一覧してから興味のある記事を選んでアクセスすることが容易に行える。コンテンツは一般的に HTML で記述された Web ページであるため、コンテンツの閲覧は RSS リーダーに内蔵された Web ブラウザまたは IE などの汎用的な Web ブラウザによってアクセスする。

RSS のサイト情報は、レジストリ[6]で探すことができる。レジストリは HTML のディレクトリインデックスに相当し、名前や分類から RSS の URL を探す。

RSS は XML で記述されており、機械的に取得して加工することが容易である。この性質を利用しているのが、コンテンツごとの情報を扱う RSS 検索エンジンとアグリゲータである。RSS 検索エンジンは、複数のサイトから定期的にコンテンツ情報を取得し、検索した結果を RSS として返す。アグリゲータ[7]は、複数のサイトから定期的にコンテンツ情報を取得し、それらを統合した RSS を公開する。

3 図書館での RSS の利用

RSS はベストセラー情報や新着情報など、更新情報を扱いやすい。

RSS を図書館に応用した例として、一橋大学附属図書館のサイト更新情報公開サービスと、今回試作した新着資料公開システムを紹介する。

3.1 サイト更新情報

一橋大学附属図書館 [8] では、図書館からのお知らせを、WWW ページで公開すると同時に、RSS として公開するサービスを開始した。RSS を使用している利用者は、自分の

[6]代表的なレジストリとして Syndic8[6] などがある。
[7]アグリゲータの例として Bulknews[2] などがある。

RSS リーダーに図書館の RSS を登録しておくことで、常に最新のお知らせを見ることができるようになった。

3.2 新着資料公開システム

図書館に到着した新着資料を RSS で配信するシステムを試作した。

簡単には新着資料のリストを RSS で公開すればよいが、次のような問題がある。

- 大学規模になると件数が多すぎる
- 個人ごとの興味に対応していない
- 複数の図書館の新着情報を統合して見たい

そこで、新着資料のリストを RSS に変換するだけでなく、独自の検索機能を持つシステムを開発し、実験を行った。

対象としたのは、大学図書館の新着資料である。新着資料の公開には、大きく 2 つの方式がある。

- 作っておいた HTML を公開する: 静的な Web ページ (A) に対応
- OPAC から日付などの条件で検索する: DB から動的に生成される Web ページ (C) に対応

新着情報は日単位で更新されることが多いので、これらを頻繁に更新される Web ページ (B) とみなし、RSS で扱うことにした。

OPAC のシステムごとに異なる形式のデータを扱うために、システム全体はラッパー・メディエータ方式で構成した。

ラッパーは、あるシステムの出力するデータ形式を標準形式に変換するプログラムである。今回は、静的な Web ページ (A) である一橋大学の新着資料ページを RSS に変換するラッパープログラムと、動的に生成される Web ページ (C) である文教大学の新着資料情報を RSS に変換するラッパープログラムを作成した。

メディエータは、複数の情報源となるデータを統合的に提示するためのプログラムである。今回は、2 種類のサイトからの新着資料情報を定期的に収集し、ユーザーごとのリクエストに合わせて検索した結果を RSS として提示するシステムを作成した。

ユーザーごとのリクエストは、大きく 2 つに分類される。

- 検索条件を指定する
- 個人を特定する

検索条件を指定する場合には、メディエータにアクセスする際に、URL の引数で検索条件を指定する。検索条件には、分類やタイトルなどを指定できる。個人を特定する場合には、ID を渡し、あらかじめ設定しておいた条件で検索する。

4 おわりに

本稿では、従来の「コンテンツを公開し、更新情報は検索エンジンの収集と反映を待つ」というモデルの限界を指摘し、続いてメタデータを公開する RSS の手法を紹介した。RSS は XML で記述されており、Dublin Core を含めて扱えるため、複数のシステム間で書誌情報を含む標準的なデータ交換フォーマットとして期待することができる。実際に図書館

または図書資料を RSS で扱った例をいくつか紹介し、実際に新着資料を RSS で扱うシステムを構築した事例を紹介した。

TP&D フォーラムの発表では、今後の活用の可能性を含めて議論を行いたい。

参考文献

[1] Atom Wiki. http://www.intertwingly.net/wiki/pie/FrontPage.

[2] Bulknews. http://bulknews.net/.

[3] RSS 0.91. http://my.netscape.com/publish/formats/rss-spec-0.91.html.

[4] RSS 1.0. http://web.resource.org/rss/1.0/spec.

[5] RSS 2.0. http://blogs.law.harvard.edu/tech/rss.

[6] Syndic8. http://www.syndic8.com/.

[7] TP&D フォーラム. http://kanemune.cc.hit-u.ac.jp/tpd/.

[8] 一橋大学附属図書館. http://www.lib.hit-u.ac.jp/service/index_Ja.html.

[9] Ward Cunningham. Wiki Wiki Web. http://c2.com/cgi/wiki?WikiWikiWeb.

[10] Ben Hammersley. *Content Syndication with RSS*. O'Reilly, 2003.

[11] Shelley Powers. *Practical RDF*. O'Reilly, 2003.

[12] 兼宗進. 検索エンジンの検索アルゴリズム. 情報の科学と技術, Vol. 54, No. 2, pp. 78–83, 2004.

[13] 愛知淑徳大学附属図書館. 司書の目と耳. http://www2.aasa.ac.jp/org/lib/j/issues_j/metomimi/metomimi.html.

[14] 神崎正英. The Web KANZAKI. http://www.kanzaki.com/.

[15] 長谷川豊祐. 図書館日記. http://www2d.biglobe.ne.jp/~st886ngw/diary/2001-1.htm.

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="0.91">
<channel>
   <title> 兼宗研究室 </title>
   <link>http://kanemune.cc.hit-u.ac.jp/tmp/rss091.xml</link>
   <description> 一橋大学 総合情報処理センター 兼宗研究室 </description>
   <copyright>Copyright (c) 2004, Susumu Kanemune</copyright>
   <language>ja</language>

<image>
   <url>http://kanemune.cc.hit-u.ac.jp/kanemune/pukiwiki.png</url>
   <link>http://kanemune.cc.hit-u.ac.jp/kanemune/</link>
   <title> 兼宗研究室 </title>
</image>

<item>
   <title> 兼宗ゼミ </title>
   <link>http://kanemune.cc.hit-u.ac.jp/zemi.html</link>
   <description> 総合情報処理センターで行っている勉強会です </description>
</item>
<item>
   <title> 論文・発表 </title>
   <link>http://kanemune.cc.hit-u.ac.jp/paper.html</link>
   <description> 論文と発表の記録です </description>
</item>

</channel>
</rss>
```

図 1: RSS0.91 の記述例

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
   xmlns="http://purl.org/rss/1.0/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xml:lang="ja">

<channel rdf:about="http://kanemune.cc.hit-u.ac.jp/tmp/rss100.xml">
   <title> 兼宗研究室 </title>
   <link>http://kanemune.cc.hit-u.ac.jp/kanemune/</link>
   <description> 一橋大学 総合情報処理センター 兼宗研究室 </description>
   <dc:date>2004-07-17T22:53:07+09:00</dc:date>
   <dc:creator>KANEMUNE Susumu</dc:creator>
   <items>
   <rdf:Seq>
   <rdf:li rdf:resource='http://kanemune.cc.hit-u.ac.jp/zemi.html'/>
   <rdf:li rdf:resource='http://kanemune.cc.hit-u.ac.jp/paper.html'/>
   </rdf:Seq>
   </items>
</channel>

<item rdf:about='http://kanemune.cc.hit-u.ac.jp/zemi.html'>
   <title> 兼宗ゼミ </title>
   <link>http://kanemune.cc.hit-u.ac.jp/zemi.html</link>
   <dc:date>2004-04-28T21:50:46+09:00</dc:date>
   <dc:subject>study, zemi</dc:subject>
   <description> 総合情報処理センターで行っている勉強会です </description>
</item>

<item rdf:about='http://kanemune.cc.hit-u.ac.jp/paper.html'>
   <title> 論文 </title>
   <link>http://kanemune.cc.hit-u.ac.jp/paper.html</link>
   <dc:date>2004-04-28T21:50:46+09:00</dc:date>
   <dc:subject>study, paper</dc:subject>
   <description> 論文と発表の記録です </description>
</item>

</rdf:RDF>
```

図 2: RSS1.0 の記述例