情報処理 (6)

WWWと検索エンジン

増原 英彦

1 World-Wide Web

1.1 歴史

ハイパーテキスト (hypertext) とは、ある文書の一部分を他の文書へ関連付け、自動的に辿れるようにしたものである。この発想自体は計算機の発明と同じ位古いものである[1]。実用的なハイパーテキストが普及したのは 1980 年代に入ってからである。現在では取扱説明書や教材のように、多くの事柄が互いに連関している文書を計算機上で見せる場合によく使われるようになっている。

一方、計算機ネットワークを通じて情報を提供・交換する仕組みは、インターネットの普及と前後して発展してきた。利用者間で文字情報をやりとりする電子メール、計算機間でファイルを転送するための FTP、文字情報を交換し多数の計算機間で電子掲示板を共有するネットワークニュースなど様々な手段が提案され、使わてきている。

World-Wide Web (WWW) は、ハイパーテキストをネットワークを越えて扱えるようにしたものと言える。文字だけでなく図を含めた情報を表示でき、マウス操作によって閲覧できるソフトウェアが一緒に作られたこともあり、急速に普及した。最初に提案されたのは 1989 年であるが、現在では多くの組織や個人による情報発信と収集の手段の一つとなっている。

1.2 仕組みの概要

WWW は、ネットワークで接続された**web サーバ**と**web ブラウザ**から成るシステムである。Web サーバは情報を提供する計算機である。Web ブラウザは、人間の指示によって web サーバから情報を取得し、表示するソフトウェアである。

1 つの web サーバは複数の情報を提供できる。ここでは提供される 1 つ 1 つの情報を「文書」と呼ぶことにする。一般に「web ページ」と呼ばれているものとほぼ同じだと考えてよい。Web ブラウザが「どの web サーバから、どの文書を」取得するかは、**URL** (uniform resource locator) によって指示する。

Web サーバが提供する文書は、主に**HTML** と呼ばれる形式をとる。この文書の中には、関連のある他の文書の URL が書き込まれている。

利用者が web ブラウザにある URL にある文書を表示するように指示すると、web ブラウザは、対応する web ブラウザと通信をして目的の文書を取得する。取得された文書が HTML 形式の情報であった場合、それを人間が見易いように整形して表示する。また、文書中に埋め込まれるべき画像などの URL が含まれていた場合は、その画像を同様にして web サーバから取得し自動的に表示する。

表示している文書中に、他の文書の URL が書き込まれていた場合、マウス操作などによって利用者からの指示に従って、その URL の文書を取得・表示する。このことを**リンクを辿る**操作と呼ぶ。

[1] Vannevar Bush, As We May Think, in *The Atlantic Monthly*, vol.176, no.1, 1945.

1.3 Web サーバ

上述のように web サーバは、ブラウザからの要求に応じて文書を返信する計算機である。単純化して説明すると、以下の 2 点に要約できる:

- web サーバには、あらかじめ提供できる文書が登録されている。それぞれの文書にはあらかじめ名前がつけられている。
- web サーバは、web ブラウザからの要求をひたすら待ち続ける。要求を受け取ると、そこに含まれている「取得したい文書の名前」に対応する文書を見つけ、その内容を送り返す。

従って提供する文書の内容については web サーバは意識をしていない。Web ブラウザからの要求に応えると言っても、それは単に予め登録されていた文書を送っているに過ぎないことに注意せよ。

今日の web サーバは上記よりも複雑なことを行うようになっている。商取引や検索などの様々なサービスを提供するためには、「あらかじめ登録しておいた文書」だけでは対応できないことは想像できるだろう。そのため、複雑な web サーバは、

- web ブラウザからの要求として、「取得したい文書の名前」[2]に加えて名前や数などの追加情報を受け取ることができ (例えば、「取得した文書の名前」として「商品の注文」を受け取る場合、追加情報として利用者の名前、商品番号、数量などを受け取り)、
- web サーバ上で動くソフトウェアがその追加情報を使って処理を行い (例えば、データベースに登録されている利用者の住所を調べ、配送センターのプリンタに注文票を印刷させる)、
- そのソフトウェアがその場で作った文書をブラウザに送る (例えば、「注文番号は○○番です」という文書を送る)

といった動作をすることができる。

1.4 HTML

WWW の文書は主に **HyperText Markup Language (HTML)** という形式で作られている。(詳細は情報発信のところで扱う。) この形式は、テキスト形式と違い、表示される文字以外に以下のような様々な情報が埋め込まれている。

- 使われている符号化形式
- 文字の大きさや色
- 箇条書きや表などの構造
- 図や他の文書を埋め込む指示
- 文章の一部を他の文書に関連付ける**リンク情報**

特に最後のリンク情報は、HTML 形式をハイパーテキストにしている重要なものである。

練習 6–1: (HTML) 自分が使っている web ブラウザで、いま見ている文書の整形される前の形式 (ソース形式) を表示させ、どこがリンク情報になっているかを推測せよ。整形される前の形式を表示するには、例えば Safari であれば「表示」メニューの「ソースを表示」を選べばよい。

[2]単純化した場合の説明に合わせて「取得したい文書の名前」と書いているが、より正確には「受けたいサービスの名前」と言うべきものである。

1.5 URL

URL (Uniform Resource Locator) は情報の場所を示す書き方である。URL は WWW に限らず、インターネット上にある場所を示す際に一般的に使われる。例えば、以下は URL の例である:

```
http://hwb.ecc.u-tokyo.ac.jp/current/
mailto:president@whitehouse.gov
rtsp://rmv8.bbc.net.uk/radio4/arts/afternoon_reading_fri.ra
```

それぞれ、WWW 文書、電子メールアドレス、動画を表わす URL である。一番目は以下のように分解できる:

《通信手順》	://	《web サーバのドメイン名》	/	《web サーバ内の文書の場所》
http	://	hwb.ecc.u-tokyo.ac.jp	/	current/

問題 6–2: (URL) 自分が使っている web ブラウザでは「現在見ている文書の URL」はどこに表示されているか?

1.6 Web ブラウザ

Web ブラウザは、web サーバから文書を取得し表示するソフトウェアである。Web ブラウザは、URL を与えられると、web サーバのドメイン名を取り出し、その web サーバに対して、サーバ内の文書を要求する。

1.7 その他

- URL の先頭にある`http` は、web サーバと web ブラウザの間の通信手順 (HTTP: HyperText Transfer Protocol) を意味している。この通信手順は、ブラウザが要求する文書の場所や追加情報を送ると、サーバが文書の種類と内容を送り返すという、比較的単純なものになっている。

  HTTP による通信は情報をそのままやりとりしているため、盗聴される可能性がある。そのため、パスワードやクレジットカードなどの情報を安全に送るために、通信内容を暗号化して送る HTTPS と呼ばれる通信手順が使われることもある。

- WWW で要求される情報の多くは、少数のものに集中する。組織単位で見ると、同じ URL の情報を何度も要求していることも少なくない。(例えば ECC の計算機からの要求を考えれば、ごく少数の「学生が日頃訪れる web サイト」に集中していることが想像できるだろう。)

  WWW プロキシ (proxy) は、このような場合にネットワークの混雑を減らすための計算機 (上で動くソフトウェア) である。WWW プロキシは web サーバへの要求を中継し、web サーバから返された文書を保管する機能を持っている。Web ブラウザが要求を送る際に、web サーバのかわりに WWW プロキシに送ると、WWW プロキシは、

  1. 過去に同じ URL への要求があったかどうかを調べ、
  2. なかった場合には、実際の web サーバに要求を送り、返信された内容を web ブラウザに送り返す。同時に、そのときの URL と返信された内容を記憶しておく。
  3. あった場合には、web サーバに要求を送らずに、記憶してあった文書を web ブラウザに送り返す。

これによって、同じ組織から同じ URL に 2 回以上要求があった場合には、2 度目以降は組織の外への通信を行うことがなくなる。これはネットワークの混雑を減らすだけでなく、web ブラウザを使っている利用者がよく早く情報を入手させる効果もある。

ただし、同じ URL の内容が時間とともに変化するような場合には、WWW プロキシは問題を起こすことがある。つまり、ある URL の内容が WWW プロキシに記憶された後で web サーバ上の情報が変更された場合、2 度目以降の要求に対して古い情報が返ってきてしまう現象である。この問題を避けるためにはいくつかの方法がとられている:

1. web ブラウザに「再読み込み」を行わせる— この場合、web ブラウザは WWW プロキシに「古い情報を使わない」指示を出す。WWW プロキシは過去に記憶した文書があったとしても、それを使わずに web サーバに要求を中継する。
2. WWW プロキシは、情報を記憶してから一定時間経過したら捨てて使わないようにする。
3. ニュースなどのように時々刻々内容が変わる情報を提供する web サーバは、要求に応えて文書を返送する際に、その文書の有効期間を付ける。WWW プロキシはその期間を過ぎた場合には、記憶していた情報を使わないようにする。

問題 6–3: (電子メールと WWW の通信の違い) 電子メールシステムと WWW はどちらも、サーバとクライアント (ブラウザ) から構成されているシステムである。以下の点に関して、両者を比較せよ:

- クライアントとサーバの間の通信は、どちらから始めるか?
- クライアントどうしは通信をするか?
- サーバどうしは通信をするか?
- クライアントは不特定のサーバと通信をするか?
- サーバは不特定の計算機と通信をするか?

2 検索エンジン

現在、世界中には数十億以上の web 文書がある。これらの文書の中から、求める情報を見つけることは容易ではない。Web サーバのドメイン名はある程度規則的に付けられているので「MIT の情報は http://www.mit.edu/ から得られるだろう」とか「IBM に関する情報は http://www.ibm.com/ から得られるだろう」といった推測はできる。しかし、よほど有名な場合以外は推測することは難しい。

検索エンジンは、名前などの手掛かりをもとに、ふさわしい情報の場所 (URL) を教えるサービスである。WWW が非常に大きな情報空間となった現在では、検索エンジンは WWW を利用するのに不可欠な存在にまでなっている。

2.1 種類

検索エンジンは、(1) 情報の探し方、(2) 情報の集め方、(3) 扱う情報の種類によって分類することができる。

1. 情報の探し方 — 文書の内容によって階層に分類された一覧表 (ディレクトリ) を人間が見て探す**ディレクトリ型**と、探したい文書に含まれていそうな言葉を使って機械的に探す**全文検索型**に大別できる。

   前者は上手く分類されている場合には効率的に情報を見つけることができる。一方後者は、分類のような作業を行わなくても、文書の中身を調べることができるので、より多くの対象を調べることができる。

2. 情報の集め方 — 人手によって集めた情報を用いる**登録型**と、ソフトウェアによって自動的に集めた文書をもとにする**ロボット型**がある。

   前者は一定の基準を満たす情報に限って調べることができるが、大量の文書を対象とすることはできない。

3. 扱う情報の種類 — 例えば料理のレシピ情報だけを検索する専用目的のものと、あらゆる WWW 文書を対象とする汎用のものがある。

以下では、Google に代表されるように、現在の検索エンジンの主流となっている、ロボット型全文検索エンジンがどのようにして検索を可能にしているかを説明する。

2.2 ロボット型全文検索エンジンのしくみ

ロボット型全文検索エンジンには、次の 3 つの段階がある:

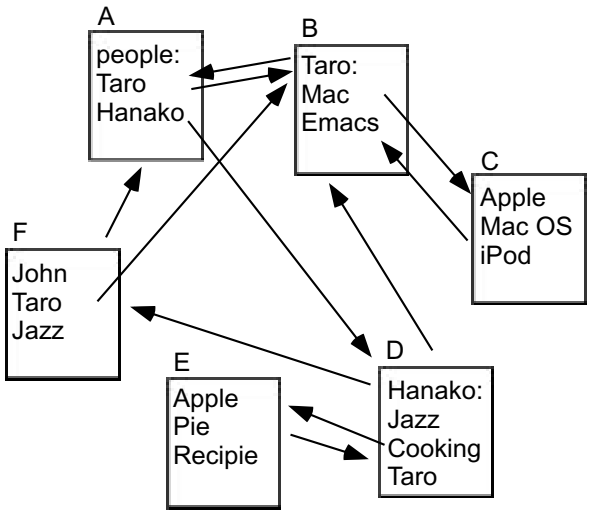
1. Web ページ収集ロボットが WWW 文書を集め、
2. 集めた文書から索引を作り、
3. 検索要求を受け付けると、索引を引いて目的の文書の URL を返す

最初の 2 段階は、検索エンジンの提供者が定期的に行っているものであり、利用者が検索エンジンを使うときは 3 の部分だけである。

2.3 Web ページ収集ロボット

検索エンジンは、**予め集めておいた WWW 文書を元に**検索をする。逆に言えば、利用者からの検索要求を受け付けてから web サーバと通信をしているのではない。

Web ページ収集ロボットは、自動的に web 文書を集めるソフトウェアである。例えば、下の図のような wbe 文書があったとする。(四角 1 つ 1 つが文書を表わし、A,B,C,... は各文書の URL だとする。文書の間に引かれた矢印はリンク情報だとする。)



ロボットはまず文書 A を取り寄せて、その URL (A) と内容を記憶する。次に、A からリンクされている B を取り寄せ、その URL と内容を記憶する。B からは A がリンクされているが、A はすでに取り寄せ済みなので無視する。そこで B からリンクされている C を取り寄せ、その URL と内容を記憶する。C からは B がリンクされているが同様に無視する。これで B から辿れる文書は全て辿ったことになるので A に戻り、A からリンクされているもう一つの URL D の文書を取り寄せて、その URL と内容を記憶する。同様にして D からのリンクを調べる、という作業を続けて文書を集めてゆく。

実際のシステムでは、数十億の文書を集めることになるので、この作業には数日あるいは数週間かかると言われている。

問題 6–4: (収集時間) 世界中に 10 億個の文書があるとして、ある収集ロボットが 10 日間でその全てを取り寄せるたとする。平均すると 1 秒間にいくつの文書を取り寄せたか?

2.4 索引付け

ページが集められたら各ページに含まれている単語を調べ、単語と URL の関係を表わす索引表を作る。上の例をもとにすれば、下のような表になる。“hanako” の列の “A” の行に 「X」があるのは、“hanako” という単語は URL A の文書に含まれていることを示している。

単語	apple	cooking	emacs	friends	hanako	ipod	jazz	john	mac	os	people	pie	recipie	taro
A					X						X			X
B			X						X					X
C	X					X			X	X				
D		X			X		X							X
E	X											X	X	
F				X			X	X						X

索引を作る意義は、検索時間の短縮である。「ある単語を含む文書」を調べるのに、取り寄せた全ての文書を 1 つ 1 つ調べていると、総文書数に比例した時間がかかってしまう。一方、索引表からその単語の列を見つけるのには総全単語数の対数に比例した時間しかからないことが知られている。単語の数は総文書数に比べると少ないため、これによって高速な検索が可能になっている。

問題 6–5: (索引を使わない検索) 索引を使わずに検索をした場合にかかる時間を見積れ。ただし、ある単語が 1 つの文書に含まれているかどうかを調べるのに、100 万分の 1 秒かかるとして、検索対象となる文書は 10 億あるとする。

2.5 検索窓口

検索は与えられたキーワードを含んでいる URL を提示するだけの作業になる。例えば「apple」というキーワードが与えられた場合、上の索引を用いて E と C の URL を提示する。この作業は、予め作られた索引を調べるだけなので、比較的効率的に行える。実際の検索エンジンでは、1 秒に満たない時間で結果を返すものが多い。

検索は基本的に「キーワードを含むかどうか」だけで URL を選び、文書の意味を調べることはない。従って「アップルコンピュータに関する情報」を調べるために「apple」というキーワードによって「アップルパイの作り方」のページが提示されてしまうこともある。

検索には複数のキーワードを使うこともできる。例えば、「apple」と「mac」の両方を含むページを見つけたければ、索引から {E, C} と {B, C} に共通して現われる C を提示する。同様にして、「どちらかのキーワードを含む文書」(例: 「cooking」か「recipe」を含む文書) や「あるキーワードを含まない文書」(例: 「apple」を含んで「mac」を含まない文書) を探すようなことも可能である。

2.6 順位付け

実世界では、一つの単語を含む文書の数は非常に多くなる。例えば「東京大学の公式ページ」を見つけるために「東京大学」をキーワードにして検索したとしても、「東京大学の学生が作ったページ」や「東京大学の入試問題を解説している予備校のページ」までも提示されてしまう。

そのため、多くの検索エンジンは、Web ページ収集ロボットが集めた文書に対して順位を付けておき、その順に従って URL を提示する。機械的に集めた億単位の文書に適切な順位を付けることは容易ではない。ここでは、Google で順位付けに使われている方法の 1 つである PageRank を紹介する。

PageRank は、各文書の価値を、その文書にリンクしている文書からのリンクの価値の和となるように定める方法である。つまり、上の例で言えば、A から F までのページの価値 $v_A, \ldots, v_F$ を、以下のような方程式の解だとする:

$$
\begin{array}{lcccccccc}
v_A &=& & \frac{1}{2}v_B & & & & +\frac{1}{2}v_F \\
v_B &=& \frac{1}{2}v_A & & +v_C & +\frac{1}{3}v_D & & +\frac{1}{2}v_F \\
v_C &=& & \frac{1}{2}v_B & & & & \\
v_D &=& \frac{1}{2}v_A & & & & +v_E & \\
v_E &=& & & & \frac{1}{3}v_D & & \\
v_F &=& & & & \frac{1}{3}v_D & &
\end{array}
$$

例えば文書 A は文書 B と D をリンクしているので、その価値 v_A は 1/2 ずつ v_B と v_D に分配されている。一方、文書 D は 2 つの文書からリンクされているので、その価値 v_D はそれぞれのリンクの価値の和になっている。

この方程式の解の 1 つは $v_A = 4, v_B = 7, v_C = 3.5, v_D = 3, v_E = 1, v_F = 1$ である。このような少数の例では、妥当性を実感することは難しいが、より沢山の文書がある場合には、順位の高い文書から沢山リンクされている文書に高い順位が付くことが分かるだろう。直感的には「人気の無い web サイトから沢山リンクされていたとしても人気の高い web サイトだとは言えない」ので、単純に「いくつの文書からリンクされているか」という順位よりも人間の感覚に近い順序で URL を提示できるようになる。

問題 6–6: (PageRank の計算) 上で示した $v_A, \ldots, v_F$ の値が方程式の解になっていることを確めよ。

問題 6–7: (実際上の問題) 上のような方法では、リンクの関係によっては解が求められない場合がある。どのような場合か? その場合には、どのように方程式を作るのがよいだろうか?

3 課題

課題 6–1 , 課題 6–2 の 2 つを 1 つのレポートにまとめ、

masuhara-js-report@lecture.ecc.u-tokyo.ac.jp

へ提出せよ。ただし、

- レポートは本文中に直接テキスト形式で書いたものを提出すること。添付ファイルによる提出や、リッチテキスト形式で書かれた提出は不可とする
- 課題を完了するのに要した時間 (作文・レポートの見本の作成を含む) と授業に対する感想も書くこと
- メールは必ず教育用計算機システムから送ること
- 自分自身のメールアドレスを Cc 欄に含めて送ること

提出期限 2004 年 5 月 31 日 (月) 23 時 59 分

提出完了の確認 期限内に提出されたレポート提出者は、締切の翌日までに授業の web ページ上にログイン名を掲示する。

課題 6–1: (情報の検索) (基本的に練習 5–9 と同じ内容である) 以下の問いについて調べ、各問いについて

- 見つけた答え
- 調べた方法 (WWW 検索エンジンによって調べた場合には、使用した検索エンジン、検索に用いたキーワードと、答えを含む URL が提示された順位などについても)
- 答えが妥当であることの根拠

を書け。

ただし、以下の点に注意せよ。(1) 調べる手段は WWW 検索エンジンに限らず、任意のものを用いてよい。(2) 答えが 1 つとは限らないような問いもある。(3) web 文書には根拠の無い情報も多いので、明確な根拠があるかどうか、2 つ以上の出典から確認できる内容であるかどうかに気を付けよ。(4) そもそも「正しい」答えがあるとは限らない問いもある。

1. イラクに現在派兵している国のリスト
2. 1997 年時点での EU 加盟国
3. マイクロソフトが過去 5 年間に提案した革新的な技術の中で、最も重要なもの 3 つ
4. ウェブログ (weblog) の意味と、その呼び方を最初に提案した人物の名前
5. 受動喫煙が胎児に与える主な影響
6. マイナスイオンが体に良い影響を与える原理

課題 6–2: (検索キーワード) 全文検索型の検索エンジンでは、検索に用いるキーワード (検索式) の与え方によって求める情報を得るまでの効率が大きく変わってくる。課題 6–1 での経験をもとに、同じ情報を求める場合でも検索式の与え方によって検索結果が大きく変わる例を示せ。さらに「どのような場合にどのような検索式を与えると効率的か」について考察せよ。