電子図書館 III

2W－8　－ Information Outlining: 触ってわかる情報の輪郭 －

丸山 宏, 諸橋 正幸, 野美山 浩
日本アイ・ビー・エム株式会社 東京基礎研究所

1　はじめに

我々が現在取り組んでいる電子図書館[1]を始め、パソコン通信、Gopher, World-Wide Web など、電子ネットワーク上で情報を提供する手段がポピュラーになり、個人がアクセスできる情報の地平線が爆発的に広がりつつある。そんな中で、欲しい情報を的確にアクセスするには、WWW サーバーなど新たな情報の固まり（これを情報集合と呼ぶことにする）に出あった時に、その集合がどのような情報を含んでいるかを自分なりに理解しておくことが非常に重要になる。

例えば、この電子図書館は近代日本文学の蔵書が豊富であるとか、このパソコンネットはパソコンの技術的な内容に非常に強いとか、ローカルな気象情報は、どの WWW サーバーが最も最新か、とか、質の高い論文のリストは、どこの ftp サイトで見つかりやすいとか、エンターテイメントで人気の高いのは、どこのＢＢＳであるか、などという自分なりの理解である。このような「情報のありかに関する知識」は、簡単に得られるものではなく、個人の経験から蓄積されたものの方が多いようである。

このように、ある情報集合の内容に関して、自分なりのイメージを描くことを、我々は、“Information Outlining” と呼ぶことにする。本稿では、“Information Outlining” が電子図書館を始め将来の情報社会に於いて非常に重要な概念であることを指摘し、これを助けるためのコンピューターの仕組みを研究する必要性を議論する。

また、その一つの方策として、検索中に、現在の検索条件に合致する文書数はいくつか、また、他の検索キーについて、その検索キーを追加すると該当文書はいくつに絞られるか、を常にユーザーに提示するアイデアについて述べる。

2　情報集合の価値

これだけ様々な情報集合が、ネットワーク上で管理されていながらも、それらすべての情報が一元管理されているわけではない。

情報で重要なのは、最新である、洩れがない (量)、内容が正確 (質)、ノイズがない、などの点であるが、例えばすべてのインターネット上の情報集合についてこのような情報を一元管理管理するのはほぼ不可能に近いことは議論の余地がないと思う。インターネット上のイエローページを作る努力はされているが、これらは上で述べたような情報の質について何ら保証するものではないし、そもそも、情報の質については、個人個人について、価値観が違うかも知れない。

本来、適切に維持管理された情報集合は、ユーザーにとって大変価値があるのだが、個人にとってどれが、自分の要求にあった情報集合であるかは、今のところ、その情報集合に実際に触れてみないとわからないものである、というのが、我々の主張である。

3　情報集合への接触

それでは、情報集合に触れるための仕組みとして必要なものは何だろうか？WWW サーバーやパソコン通信などの情報集合を考えてみると、通常提供されている手段は、階層構造方式のメニューや、キーワード検索によるものがほとんどである。

メニュー方式は一番良く使われる。階層式のメニューは、メニューをうまく作れば、一見しただけでこの情報集合がどんな内容を含んでいるかをかなり表現できる。例えばよく整理されたディレクトリの構造があれば、欲しいファイルに到達するのは容易である。しかしながら、メニューは、そのメニュー項目の内容を、簡単な記述でしか表すことができない。例えば、あるディレクトリの下に、いったい何バイトのデータがあるのかは、実際に階層構造を下ってみないとわからない。したがって、魅力的なメニュー項目に誘われてそのメニューの下をさまよい歩いても、実際にはあまり役に立つ情報がなかったりするのである。

キーワード検索は、「自分が欲しい情報が何であるか」がはっきりわかっている時に、特に、「自分の欲しい情報が確かにここにある」とわかっている時に非常に役に立つ。一方、「この情報集合にはどんな情報があるのだろう」とウィンドウ・ショッピングのようにさまよい歩くにはあまり役に立たない。キーワード検索で出てくる情報は、あくまでもユーザーが思いついたキーワードという断片で情報集合を切りとっただけのものであり、ユーザーが思いもよらないところに大量の情報を持っているかもしれないからである。

このように、現在のところ、「集合としての情報」に触れてみるための道具立てが、あまり発達していないようである。コンピュータネットワークが発達した現在、情報集合の個人的な評価を助けるツールがこれからは重要になるであろう。

4　評価を助けるツール

個人にとっての情報集合の価値を推定するのは簡単ではないと思われるが、情報集合の、カバレージや最新性については、ある程度統計量を表示することによって助けることができると思われる。ここでは、そのような目的に沿って考えられた一つのアイディアを紹介する。

前稿で述べたように、我々は情報集合に対して、複数のビューから階層構造のメニューを持つことを考えている[1]。ここで述べ

Information Outlining: Touch and Find the Profile of Information
IBM Research, Tokyo Research laboratory
Hiroshi Maruyama, Masayuki Morohashi, Hiroshi Nomiyama

るアイディアは、これらのビューの中のそれぞれのメニュー項目に、「この項目を選ぶと該当する文書はいくつになるか」を計算して表示しておくことである。当然、あるビューにおいて一つの項目を選ぶと、他のビューの該当文書数が全部再計算されて表示される。ビューは基本的にはメニューだが、地図やグラフのようなビジュアルなものでも構わない。大事なことは、ユーザーがある項目を選ぶと、すべての数字が瞬時にアップデートされ、それによって、この情報集合に入っている情報の傾向が、(少なくとも量の観点からは) ユーザーに一目瞭然にすることである。

ところが、我々が想定している電子図書館などでは、対象文書が膨大であるため、現行のビューの組合せに合致する文書数を、「それらの文書を数え上げることによってリアルタイムに求める」ことはほとんど不可能である。そこで、以下のような手段で、メニュー項目の組合せに該当する文書数を推定することにする。以下の議論では、メニュー項目をキーワードと呼ぶことにする。

本手法の大筋は以下のようである。

1. すべての検索キー x について、その x を含む文書数 $f(x)$ をあらかじめ計算しておく。これは、キーワード集合の大きさを N とすれば、N に比例する領域で記憶できる。

2. すべての検索キーのペア $\langle x, y \rangle$ について、x と y を同時に含む文書数 $f(xy)$ をあらかじめ計算しておく。これは、N^2 の領域で記憶できる。

3. 任意の検索キーの組合せ $x_1, x_2, ..., x_m$ について、それらすべてを同時に含む文書数 $f(x_1x_2...x_m)$ を、$f(x_i)$ および $f(x_ix_j)$ (但し、$1 \leq i \leq m$, $1 \leq j \leq m$, $i \neq j$) を用いて推定する。

上記 1,2 には、あわせて N^2 に比例する記憶域が必要であるが、3 において、文書数に依存しない時間で $f(x_1x_2...x_m)$ を推定できる。

キーワード集合の大きさ N はビューとそのメニュー階層に依存するが、数千、あるいは数万のオーダーになると考えられる。これをナイーブに実現すると数 10MB〜数 100MB の領域が必要となる。だが、実際には、非常にスパースなマトリクスになること、及び、たとえ数 10MB〜数 100MB としても、文書全体のサイズに比べればたいして大きくないことから、実用上の問題はないであろう。

4.1 3項での推定

例題として、x、y、z の 3 つの検索キーを考えよう。それぞれを含む文書数を $f(x)$、$f(y)$、$f(z)$、また 2 つの検索キーを含む文書数をそれぞれ、$f(xy)$、$f(yz)$、$f(zx)$ とする。これらの情報だけで、3 つの検索キーを含む文書数 $f(xyz)$ を正確に求めることはできない (例として、$f(x) = f(y) = f(z) = 100$、$f(xy) = f(yz) = f(zx) = 30$ の場合を考えてみると良い。$f(xyz)$ は、0 から 30 までの任意の値を取り得るので、正確な数は計算できない) が、以下の仮定を置くことによって、推定することができる。

仮定 1

キーワード x, y を持つ文書のうち、キーワード x, y, z を持つ文書の割合は、キーワード x を持つ文書が、キーワード x, z を持つ割合と等しい。(すなわち、キーワード x を持つ文書がキーワード z を持つかどうかは、それがキーワード y を持つかどうかとは独立であるとする。)

この仮定を置くと、$f(xyz)$ の値は、次の 3 通りの推定ができる。

$$f(xyz) = \begin{cases} f(xz)f(yz)/f(z) \\ f(xy)f(yz)/f(y) \\ f(xy)f(xz)/f(x) \end{cases}$$

これらの値は一般には互いに異なるため、全体を勘案して推定を行なう、ならし関数 M を用意して、

$$f(xyz) = M(f(xz)f(yz)/f(z), f(xy)f(yz)/f(y), f(xy)f(xz)/f(x))$$

のようにする。M としては、単純には相加平均あるいは幾何平均のようなものでも構わない。

但し、いくつかの特殊条件は考慮に入れる必要がある。例えば、

$$f(xyz) \leq \min(f(xz), f(yz), f(xy))$$

であるし、また、$f(xy) = f(y)$ である時には、$f(xyz) = f(yz)$ となる。

4.2 m 項関係への拡張

3 項で推定ができれば、m 項関係に拡張するのは容易である。キーワード $x_1, x_2, ..., x_m$ が与えられた時に、それらの全てを含む文書の数 $f(x_1x_2...x_m)$ は、以下の式で推定する (M として相加平均を仮定、上述の特殊条件については省略)。

$$f(x_1x_2...x_m) = \frac{2}{m(m-1)(m-2)} \sum_{i=1}^{m} \sum_{j=1}^{m,i\neq j} \sum_{k=j+1}^{m,i\neq k} \frac{f(x_ix_j)f(x_ix_k)}{f(x_i)}$$

このようにすると、m 個の検索キーが与えられた時に、$f(x_1x_2...x_m)$ を求めるためには、$O(m^3)$ の計算量で計算でき、また、この際に必要なデータベースアクセスは、$m(m+1)$ 回で済む。これは全体の文書数によらない。したがって、高速にリアルタイムに計算することができるのである。

5 おわりに

我々は、この手法を、電子図書館ナビゲーションプロトタイプ MELSINE の一機能として実現した。日経新聞一年分のデータに対して、地理、時間、分野などのビューを用意し、これらの組合せの該当記事数を本手法で計算している。計算はほとんど瞬時であり、また、推定と実際の該当記事数も大きく違わないことを確認している。

参考文献

[1] 堤、諸橋、丸山、他 1994: "電子図書館 I" 情報処理学会全国大会予稿集.