
インターフェイスの街角 (68)– Web ページ用の索引

増井 俊之

「索引のない本など本ではない」と言われることがあります。たしかに、最近の計算機関係の専門書では、索引のないものはほとんどないようです。ところが、日本ではどちらかといえば索引のない書籍のほうが多く、自然科学分野の翻訳書などの場合でも、原著には立派な索引があるのに訳書では完全になくなっていて困ることがよくあります。

最近は全文検索システムがひろく使われるようになってきたため、Web ページや計算機上の文書を検索するぶんには、それほど不便を感じることはありません。しかし、普通の全文検索システムでは、対象となる文書のなかで使われている文字列と同じキーワードを入力しないと検索に成功しないので、類語や漠然とした概念を指定しただけでは目指す箇所にとどり着けません。

Google などの検索エンジンでは、特定の単語がインターネット上でどのように使われているかを詳しく調べることはできます。しかし、明確な言葉で表現しにくい概念を検索しようとする、検索キーワードの選択に苦労します。たとえば、「日本の書籍における索引の現状」について調べようとする場合、

- 書籍 索引 現状
- 索引の現状
- 索引の問題

といったキーワードを指定しても、書籍の索引の現状についてのページはほとんどヒットしません。一方、「索引のない本」というキーワードで検索すると、「索引のない本など本とは呼べない」とか、「索引のない本が多いのはまったく困ったものだ」のように、索引のない書籍の問題について書かれたページがいくつかみつかります。このような Web ページの説明文に「日本の書籍の索引の現状」の

ような文字列が書いてあったなら、上記の「書籍 索引 現状」のようなキーワードでも検索できたはずですが。

学術論文では、冒頭に概要 (アブストラクト) やキーワードを列挙するのが通例となっていて、著者が指定したキーワードが索引として利用されることもあります。それでも、使用するキーワードは著者ごとに異なっていたりするので、実際には検索に有効に活用されることはあまり多くないようです。

インターネットなどで情報を広めたいときは、タイトルや全文検索だけを利用するのではなく、可能なかぎり多様な表現で検索できる索引を用意しておくとは有効な場合があるのではないのでしょうか。

充実した索引の生成

たとえば、Emacs でカーソルを動かす方法に関する資料の索引を作るとき、その章のタイトルは「カーソル移動」かもしれませんが、あるいは「カーソルを動かす」かもしれません。いずれにせよ、カーソルの動かし方を知りたいと思ったユーザーが「カーソル」「移動」「動かす」といったキーワードのうちのどれを用いて検索するかは分からないので、考えうるすべてのキーワードを索引に入れておくのが得策でしょう。

TeX でこのような展開索引を作りたいときは、make-index の機能を使います。TeX のファイルに以下のようなコマンドを埋め込んでおくと、索引情報が索引用補助ファイル (*.idx) に出力されます。

```
\usepackage{makeidx}
\makeindex
.....
\section{カーソル移動}
```

```
\index{カーソルを移動する}
\index{移動する!カーソル}
\index{動かす!カーソル}
```

生成された idx ファイルを makeindex コマンドで処理すると、以下のような形式の索引が得られます。

```
\begin{theindex}
  \item カーソルを移動する, 1
  \indexspace
  \item 移動する
    \subitem カーソル, 1
  \indexspace
  \item 動かす
    \subitem カーソル, 1
\end{theindex}
```

このように、TeX では makeindex の基本機能を使って索引データを自動生成することができます。とはいえ、いろいろなキーワードや表現を用いて索引を充実させるのは、やはりかなりの大仕事です。

Permuted Index

Emacs Lisp のマニュアルなどでは、索引に入れる項目を部分文字列でソートした Permuted Index というものがよく使われています。たとえば very-long-function という名前の関数についての索引であれば、

- very-long-function
- long-function, very-
- function, very-long-

のように分解し、それぞれを索引の v、l、f の場所に配置しておきます。そうすれば、単語の途中の“long”しか思い出せなくても検索に成功するというわけです。

同様の手法を使えば、“カーソルを動かす”のような表現から“カーソル”“動かす、カーソル”といった索引項目を自動的に作成できるでしょう。

展開索引の生成

2002 年 2 月号で、検索対象となるテキストをいろいろな表現に展開したものに対して検索をおこなう手法について解説しました。索引を作成する場合も、索引項目を多様な表現に展開しておけば、より充実した索引が得られるはずです。

たとえば、“カーソルを動かす”に索引をつける場合、

図 1 Emacs のカーソル移動の説明ページ



- “~を動かす”は“~を移動する”と同義であるという言換え辞書を用意し、“カーソルを移動する”のようなテキストを生成する
- “動かす、カーソル”“移動する、カーソル”のような Permuted Index を生成する

という処理をおこなえば、それほど手間をかけずに緻密な索引が作れます。

Web ページの索引

最近の大規模な Web サイトでは、フレームなどの機能を使って、書籍の目次に相当する部分がつねに画面の上段や左右の枠内に表示されるように構成されたページをよくみかけます。しかし、書籍の索引に相当するページを用意してあるサイトはあまりみかけません。

全文検索機能があるから索引など不要だという考えなのかもしれませんが、前述のように全文検索は万能ではありません。よくできた索引を用意しておけば、全文検索よりも役立つことがあるはずです。

一例として、図 1 のような Web ページに索引をつけることを考えてみましょう。

図 2 のソースを見れば分かるように、このページには“移動”という単語はありません。したがって、ユーザーがカーソルを動かす方法を調べようとして“移動”をキーワードにして全文検索を実行しても、このページはみつかりません。しかし、索引に“移動”というエントリがあり、このページへのリンクが張られていれば、全文検索をおこなうまでもなく、“移動”を手掛かりにこのページを見つけることができるでしょう。

図 2 Emacs のカーソル移動の説明ページのソース (cursor.html)

```
<html>
<head>
<title>Emacsのカーソル操作</title>
</head>
<body>
<h1>Emacsのカーソル操作</h1>
本資料ではEmacsのカーソル操作について説明します。
<h2><a name="ud">カーソルを上下に動かす</a></h2>
画面上のカーソルを1行上に動かすにはCtrl-Pキーを、下に動かすにはCtrl-Nキーを使用します。
<h2><a name="lr">カーソルを左右に動かす</a></h2>
画面上のカーソルを1文字左に動かすにはCtrl-Bキーを、右に動かすにはCtrl-Fキーを使用します。
<h2><a name="line">カーソルを行の左端/右端に動かす</a></h2>
画面上のカーソルを行の先頭に動かすにはCtrl-Aキーを、右端に動かすにはCtrl-Eキーを使用します。
.....
```

図 3 Emacs 操作の索引テキスト

cursor.html	[カーソル]の[操作]
cursor.html	[カーソル]を[動かす]
cursor.html#ud	[カーソル]を[上下]に[動かす]
cursor.html#ud	[カーソル]を[上]に[動かす]
cursor.html#ud	[カーソル]を[下]に[動かす]
cursor.html#ud	[カーソル]を[次]の[行]に[動かす]
cursor.html#ud	[カーソル]を[前]の[行]に[動かす]
cursor.html#lr	[カーソル]を[左右]に[動かす]
cursor.html#lr	[カーソル]を[左]に[動かす]
cursor.html#lr	[カーソル]を[右]に[動かす]
cursor.html#line	[カーソル]を[行頭]に[動かす]
cursor.html#line	[カーソル]を[行(ぎょう)]の[先頭]に[動かす]
cursor.html#line	[カーソル]を[行(ぎょう)]の[左端]に[動かす]
cursor.html#line	[カーソル]を[行末]に[動かす]
cursor.html#line	[カーソル]を[行(ぎょう)]の[最後]に[動かす]
cursor.html#line	[カーソル]を[行(ぎょう)]の最[後尾]に[動かす]
cursor.html#line	[カーソル]を[行(ぎょう)]の[右端]に[動かす]
cursor.html#para	[カーソル]を[次][パラグラフ]に[動かす]
cursor.html#para	[カーソル]を[前][パラグラフ]に[動かす]
cursor.html#word	[カーソル]を[次]の[単語]に[動かす]
cursor.html#word	[カーソル]を[前]の[単語]に[動かす]
cursor.html#blink	[カーソル]の[点滅]を[制御]
cursor.html#color	[カーソル]に[色(いろ)]をつける

索引テキストの準備

単語の置換えや Permuted Index の処理が簡単におこなえるように、キーワードの区切りを明示したテキストを索引に使用することにします。

さきほどの図 1 の Web ページには、`カーソルを上下に動かす`という項目名と`カーソルを 1 行上に動かす`のような説明文が書いてあります。これらに加えて、`カーソルを上を移動` `カーソルを次の行に移動`といったテキストも別表現として記述しておきます。

図 3 のテキストに対して、図 4 の単語置換え辞書を適用して展開すると、`カーソルを上下に移動する`のようなテキストが得られます。また、角括弧の部分で区切って

図 4 単語置換え辞書

動かす	移動
パラグラフ	段落
.....

Permuted Index を生成すれば、

- カーソルを上下に動かす
- 上下に動かす, カーソルを
- 動かす, カーソルを上下に
- カーソルを上下に移動する
- 上下に移動する, カーソルを
- 移動する, カーソルを上下に

図 5 展開されて作成された索引



図 6 同じ意味のエントリを削ってすっきりさせる



図 7 QuickML の索引テキスト

```
usage.html#create [メーリングリスト]を[新たに][作成する]
usage.html#create [新しい][メーリングリスト]を[作成する]
usage.html#create [メーリングリスト]を[新規]に[作成する]
usage.html#submit [メーリングリスト]へ[メッセージ]を[投稿]する
usage.html#submit [自分]が[参加]している[メーリングリスト]へ[メッセージ]を[送る]
usage.html#submit [自分]が[参加]している[メーリングリスト]へ[メッセージ]を[送信]する
.....
```

のようなエントリが得られます。これらをすべて索引項目としておくと、“上下”“カーソル”“移動”“動かす”など、いろいろなキーワードによる検索が可能になります。

この結果をソートして索引らしく整形すると、図 5 のようになります。もとの索引テキストは 23 行の短いものでしたが、163 項目もの索引に展開されました。

単純に単語を置換して Permuted Index を作成すると、似たようなエントリが並んでしまう場合があります。たとえば、図 5 では、

- カーソルを下に移動
- カーソルを下に動かす
- カーソルを次段落に移動
- カーソルを次段落に動かす

といったエントリが連続して並び、かなりくどい感じがします。同じ意味のエントリが並んだ場合は省略して表示すれば、もうすこし見通しがよくなるでしょう。

このような処理を加えると索引項目の数は 87 になり、図 6 のようにすっきりします。

実サイトへの適用

この手法を応用して、お手軽メーリングリスト作成サービス QuickML.com¹の索引を作成してみました。

まず、索引テキストとして図 7 のようなものを用意します。

このテキストを展開して索引を作成すると、図 8~9 のような索引ページが生成されます。

47 項目の解説へのリンクが、図 7 では 157 行の説明文で表現され、これが図 8~9 では 392 件の索引項目に展開されています。単純計算でいえば、普通に索引を作った場合にくらべて 8 倍ほど詳しい索引が得られたこととなります。

ただし、このままだと索引ページがかなり長くなってしまっているので、索引として出現するすべてのキーワードを図 10 のようにまとめてもよいでしょう。こうしておけば、全文検索システムでキーワードを入力する代わりに、キーワードをマウスでクリックするだけで検索できます。さらに、

1 <http://QuickML.com/>

図 8 QuickML の索引(先頭)



文書に含まれていない単語も検索に使えるので、結果として強力な検索が可能になります。

おわりに

索引を真剣に作成しようとなると、膨大な手間がかかります(欧米などでは、Indexer という索引作成専門の職業があるほどです)

図 7 のように、QuickML の FAQ に対していろいろな言換え表現を用いたページを用意し、かなり広範な表現を網羅できたと自負していたのですが、用意した検索キーワードにまったくマッチしない質問をユーザーから受け取ってがっかりしてしまいました。広範な検索要求に応えられるキーワードを準備するのは、やはりそう簡単なことではないようです。

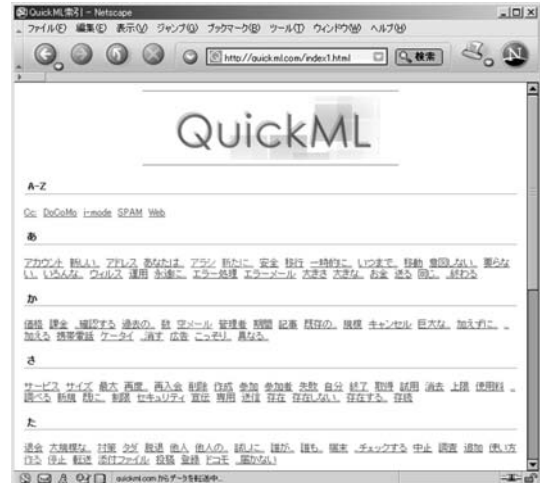
しかし、丁寧に作業をすれば、検索要求の 8 割くらいは満たせる索引を作るのはそれほど難しくないのであります。用意したものにマッチしない検索要求を受け取ったときは、それをそのまま索引として新たに登録してしまえばいいわけですから、索引の保守は容易だと思います。

ある程度よく使われるシステムについては、今回のような方法で丁寧に索引を作り、Web 上に索引ページを作っておくことは意義があるでしょう。関連するいろいろなキーワードを含む索引ページを作っておけば、そこに記した項目も Google などの検索対象になるので、全文検索に対

図 9 QuickML の索引(末尾)



図 10 索引キーワードのリスト



しても有効です。全文検索システムはたいへん便利なものですが、それだけに頼らず、適切な索引ページも用意しておく、さらに検索効率上がるのではないのでしょうか。

今回の索引生成システムは、私の Web ページ²で公開しています。ぜひご利用ください。

(ますい・としゆき 産業技術総合研究所)

2 <http://pitecan.com/PermutedIndex/>