ITR5 Information Usage

Lecture 8

Thomas Krichel

2002–03–06

Reading

Paul J. Lucas' swish++ homepage at

http://homepage.mac.com/pauljlucas/software/swish/

Source code of swish++

Brin, Sergey and Page, Lawrence "The Anatomy of a Large-Scale Hypertextual Web Search Engine", availabable at http://dbpubs.stanford.edu:8090/pub/1998-8

Page, Lawrence, Sergey Brin, Rajeev Motwani and Terry Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", available at http://citeseer.nj.nec.com/page98pagerank.html

index format (simplified explanation)

word data

where word is the word and *data* is one or more structures of the form

file number_of_occurences rank

where *file* has data about a file *number_of_occurences* is the number of occurences in that file and *rank* is the rank of the word in the file.

(I leave out discussion of meta terms to simplify.)

Queries without Meta Data

The query

```
librar*
```

will return all documents that contain "library," "libraries," or "librarian." The query:

```
mouse and computer
```

will return only those documents regarding the kind of mice attached to a computer and not the rodents. The query:

```
cat or kitten or feline
```

will return only those documents regarding cats. The query:

```
mouse or mice and not computer
```

will return only those documents regarding mice (the rodents) and not the kind attached to a computer. The query:

```
mouse and computer or keyboard
```

is the same as:

```
(mouse and computer) or keyboard
```

in that they will both return only those documents regard ing either mice attached to a computer or any kind of key board. However, neither of those is the same as:

```
mouse and (computer or keyboard)
```

that will return only those documents regarding mice and either a computer or a keyboard.

Queries Using Meta Data

The query:

```
author = carroll
```

will return only those documents whose author attribute contains "carroll." The query:

```
author = stevenson treasure
```

will return only those documents whose author attribute contains "stevenson" and also regarding treasure. The query:

```
author = (lewis carroll)
```

will return only those documents whose author is Lewis Carroll. The query:

```
author = (lewis carroll) or wonderland
```

will return only those documents whose author is Lewis Carroll or that contain the word "wonderland" anywhere in the document regardless of the author.

Compute the rank of a word in a file.

"This equation was taken from the one used in SWISH-E whose author thinks (?) it is the one taken from WAIS. I can't find this equation in the refernece cited below, although that reference does list a different equation. But, if it ain't broke, don't fix it."

rank = (log (occurences in file) + 10) * 10000 / (total number of occurences of word in all files) / (total words in file).

Gerard Salton. "Automatic Text Processing: the transformation, analysis, and retrieval of information by computer." Addison-Wesley, Reading, MA. pp. 279–280.

The "Clinton sucks" problem

Let the query be "Bill Clinton" and the page say

"Bill Clinton sucks" and have a picture.

Page highly related, but poor quality.

Page quality

If we index the whole of the web, and we search for terms on that huge collection, we need a way to select high-quality pages, and display those first.

How can that be done?

What is an important page

A page that many pages point to is important.

A page that is pointed to by other important pages is important.

Model of user behavior

A random surfer is given a page.

S/he keeps clicking on any link that she finds in that page with a probability d.

S/he gets bored and starts with a completely new random page with probability $1 - d$.

Does that surfer hit all pages on the web with the same probability?

Page Rank.

Let there be a page A. Let $l(A)$ be the number of links that go out of A. Let there be a number of pages P_1,...,P_n that have links to A. Then the of A, note $r(A)$ is given by

$$r(A) = \epsilon_A (1 - d) + d \left(\frac{r(P_1)}{l(P_1)} + \frac{r(P_2)}{l(P_2)} + \ldots + \frac{r(P_n)}{l(P_n)} \right)$$

where $1 - d < 1$ is the probabilty that the user gets bored, and $\epsilon_A > 0$ is the probability, that if the user is bored, (s)he picks page A. Of course $\sum_{\forall P} \epsilon_P = 1$.

Interpret that formula.

Example

Imagine the web is composed of four pages A, B, C, D.

A links to B, C, D.

B links to A.

C links to B and D.

D links to B.

Calculate the page rank for each page, assuming that $1-d = 1/4$ and $\epsilon = 1/4$ for all pages.

Other features

- proximity is search terms
- uses visual representation data i.e. bigger print is more important than small print
- usage of link anchor data