

## Retrieving Information from the Web

Database and Information Retrieval (IR) Systems both manage data !

- The data of an IR system is a *collection of documents* (or *pages*)

User tasks:

- ◇ *Browsing* - examining documents
- ◇ *Retrieval* - searching for documents

### Database System versus IR System

Category	SQL	Search Engine
<i>Matching</i>	exact answer	ranked answer
<i>Language</i>	sophisticated	simple
<i>Algorithm</i>	deterministic	probabilistic
<i>Database</i>	structured	semistructured
<i>Query</i>	complete	incomplete
<i>Error</i>	sensitive	insensitive

Home Page

Title Page

◀ ▶

◀ ▶

Page 2 of 14

Go Back

Full Screen

Close

Quit

## The Web is a Hypertext System

- *Content* - collection of pages
- *Structure* - links (directed graph)

Additional user task:

◇ *Navigation* - traversing links and following a *trail* of associated links

Quote from Bush 1945 “As We May Think” (download from my web links):

“the process of tying two items together is an important thing .. when numerous items have been thus joined together to form a trail they can be reviewed in turn”

Nelson’s vision of a *universal hypertext database* - *Xanadu* (1960’s)

[Home Page](#)

[Title Page](#)



Page 3 of 14

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## The Basic Information Retrieval Algorithm

1. Remove stopwords such as: of, the, a ...

2. Apply stemming to *terms* (or words),

i.e. remove prefixes and suffixes

E.g. connected, connecting, connection and connections  $\Rightarrow$  connect

3. Weight the terms in the query and in pages

4. Rank the pages according to similarity with the query

## Term Weighting

$N$  - total no. of pages in the system

$n_j$  - no. pages in which term  $j$  appears

### term frequency

$tf_{ij}$  = frequency of term  $j$  in page  $i$

### inverse document frequency

$$idf_j = -\log \frac{n_j}{N} = \log \frac{N}{n_j}$$

(self-information of term  $j$ )

### normalised term weight

$$w_{ij} = \frac{tf_{ij} \times idf_j}{\max_k tf_{ik}}$$

### Query Weighting

$$w_{qj} = \left( 0.5 + \frac{0.5 \times tf_{qj}}{\max_k tf_{qk}} \right) \times idf_j$$

[Home Page](#)[Title Page](#)[Page 5 of 14](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

## Similarity

$m$  - no. of terms considered

Represent page  $i$  as a vector

$$\mathbf{i} = \langle w_{i1}, w_{i2}, \dots, w_{im} \rangle$$

Represent query  $q$  as a vector

$$\mathbf{q} = \langle w_{q1}, w_{q2}, \dots, w_{qm} \rangle$$

$$\text{sim}(i, q) = \sum_{k=1}^m w_{ik} \times w_{qk}$$

(dot product of  $\mathbf{i}$  and  $\mathbf{q}$ )

(Other similarity measures exist)

Home Page

Title Page

◀ ▶

◀ ▶

Page 6 of 14

Go Back

Full Screen

Close

Quit

## Measures of Information Retrieval

$R_F$  - no. relevant pages returned

$R_N$  - no. relevant pages *not* returned

$I_F$  - no. of irrelevant pages returned

$I_N$  - no. of irrelevant pages *not* returned

$$\text{recall} = \frac{R_F}{R_F + R_N}$$

- Proportion of relevant pages returned.

$$\text{precision} = \frac{R_F}{R_F + I_F}$$

- Proportion of returned pages which are relevant.

◇ Precision versus Recall curve

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 7 of 14](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

## Searching the Web

- ♣ over 2 billion pages (2000) growing at 1 million pages per-day.
- ♣ Each page has on average 7 out-links.
- ♣ Over 600 GB of text changes every month.
- ♣ Largest crawlers cover 30-40% of the indexable web during several months.
- ♣ 10 percent redundancy in mirrored sites.
- ♠ Most users type in short queries on average less than 3 terms.
- ♠ Most users only look at the top ten results.
- ♠ Most users do not modify their original query.

[Home Page](#)[Title Page](#)[◀](#) [▶](#)[◀](#) [▶](#)[Page 8 of 14](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

## Using Link Structure in Search

$L_{ij} = 1$  if there is a link from  $i$  to  $j$  and 0 otherwise.

### Structured Weighting

$$sw_{qj} = w_{qj} + \sum_{k \neq j} \alpha L_{kj} \times w_{qk}$$

- $\alpha$  is between 0 and 1 (0.2 seems to be optimal)
- the sum is over all pages that have a link to page  $j$

Home Page

Title Page



Page 9 of 14

Go Back

Full Screen

Close

Quit

## HITS - Hypertext Induced Topic Search

Given a query such as “XML” distinguish between:

- **authorities** - pages which focus on the topic of XML such as various publications on the XML standard.
- **hubs** - pages that contain many useful links to relevant pages
- A densely linked focused subgraph of hubs and authorities is called a *community*.
- Over 100,000 emerging web communities have been discovered from a web crawl (a process called *trawling*).

## The HITS Algorithm

1. Collect the top  $t$  pages (say  $t = 200$ ) based on similarity, called the *root set*.
2. Extend the root set into a *base set* as follows, for all pages  $p$  in the root set:
  - 2.1. add to the root set all pages that  $p$  points to, and
  - 2.2. add to the root set up-to  $d$  pages that point to  $p$  (say  $d = 50$ ).
3. Delete all links between the same web site in the base set resulting in a *focused subgraph*.
4. Assign to each page  $p$  a non-negative *hub weight*  $y_p$  and a non-negative *authority weight*  $x_p$ .
5. Iteratively reinforce hubs and authorities as follows, until convergence:

$$x_p := \sum_q \text{ where } q \rightarrow p y_q$$
$$y_p := \sum_q \text{ where } p \rightarrow q x_q$$

## PageRank - Google

Model of a “random surfer”:

1. The surfer given a web page at random.
2. The surfer follows “forward” links without going “back”.
3. When the surfer gets bored a random page is chosen as the next page.

- The PageRank of a page is the probability that a random surfer visit a page

$P$  - a page which has incoming links from pages  $P_1, P_2, \dots, P_n$

$r$  - a positive number between 0 and 1

$O(P_i)$  - the number of links going out of page  $P_i$

$$PR(P) = r + (1 - r) \sum_{i=1}^n \frac{PR(P_i)}{O(P_i)}$$

Home Page

Title Page

◀ ▶

◀ ▶

Page 12 of 14

Go Back

Full Screen

Close

Quit

## Metasearch

**Problem:** Search engines have limited coverage and overlap (Nature 1999)

- ♣ Relative coverage of major search engines about 20%.
- ♣ The overall coverage is small, less than 16% are indexed by all engines, not taking into account the *deep web*.

**Solution:** Select and merge results from several data sources

- Not easy to do well due to heterogeneity of local search engines

Home Page

Title Page



Page 13 of 14

Go Back

Full Screen

Close

Quit

## The Navigation Problem in Hypertext

The steps in searching for information:

- 1) **Query** - user provides the context
- 2) **Information Retrieval** - ranked list of pages returned
- 3) **Navigation** - user *repeats* :
  - (a) choose a page to *browse*
  - (b) follow a *link*
- 4) **Query Modification** - user *returns* to (1)

Home Page

Title Page



Page 14 of 14

Go Back

Full Screen

Close

Quit

**Problem:** “getting lost in hyperspace” - navigation (link following) leads to disorientation in terms of the *goals* and *relevance* of the currently browsed page to the query.

**Solution:** Trails are first-class citizens

- We develop algorithms which maximise the expected trail relevance.