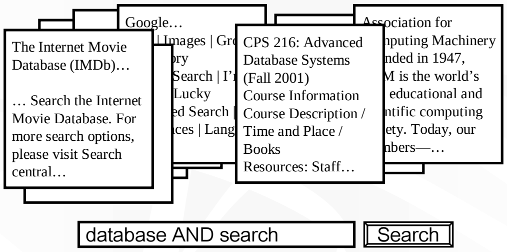
Even More Indexing!

CPS 216

Advanced Database Systems

Keyword search



Google…
| Images | Gro
ory
Search | I'r
Lucky
ed Search |
ces | Lang

The Internet Movie Database (IMDb)…

… Search the Internet Movie Database. For more search options, please visit Search central…

CPS 216: Advanced Database Systems (Fall 2001)
Course Information
Course Description / Time and Place / Books
Resources: Staff…

Association for
nputing Machinery
nded in 1947,
M is the world's
educational and
ntific computing
ety. Today, our
nbers—…

What are the documents containing both "database" and "search"?

Search features

- Boolean searches
  - (database OR Web) AND search
- Phrase searches
  - "database search"
- Result ranking
  - Number of occurrences of keywords in the document
  - Proximity of keywords within the document
  - Popularity of document
  - Google, Teoma, etc., etc.

Keywords × documents

All documents

All keywords	Document 1	Document 2	Document 3		Document *n*
"a"	1	1	1	…	1
"cat"	1	1	0	…	0
"database"	0	0	1	…	0
"dog"	0	1	0	…	1
"search"	0	0	1	…	0
	…	…	…	…	…

1 means keyword appears in the document
0 means otherwise

Inverted lists

- Store the matrix by rows
- For each keyword, store an inverted list
  - <*keyword*, *document-id-list*>
  - <"database", {3, 7, 142, 857, …}>
  - <"search", {3, 9, 192, 512, …}>
  - It helps to sort *document-id-list* (why?)
- Vocabulary index on keywords
  - B+-tree or hash-based

Using inverted lists

- Documents containing "database"
  - Use the vocabulary index to find the inverted list for "database"
  - Return documents in the inverted list
- Documents containing "database" AND "search"
  - Return documents in the intersection of the two inverted lists
    - It helps to keep inverted lists sorted!
- OR? NOT?
  - Union and difference, respectively

What are “all” the keywords?

- All sequences of letters?
  - … that actually appear in documents!
- All words in English?
- Plus all phrases?
  - Alternative: approximate phrase search by proximity
- Minus all stop words
  - They appear in nearly every document; not useful in search
  - Example: a, of, the, it
- Combine words with common stems
  - They can be treated as the same for the purpose of search
  - Example: database, databases

Frequency and proximity

- Frequency
  - <*keyword*, {<*doc-id*, *number-of-occurrences*>,
    <*doc-id*, *number-of-occurrences*>,
    … }>
- Proximity (and frequency)
  - <*keyword*, {<*doc-id*, <*position-of-occurrence*$_1$,
    position-of-occurrence$_2$, …>,
    <*doc-id*, <*position-of-occurrnece*$_1$, …>,
    … }>
  - When doing AND, check for positions that are near

Ranking Web pages using links

- Basic idea: A page is relevant if a lot of relevant pages have links pointing to it
  - Recursive definition?
    - No problem—fixed-point iteration!
- Google
  - Pre-compute the “general” ranking of all pages
  - This ranking can be use in the inverted lists
- HITS, Teoma
  - Compute the “topic-specific” ranking dynamically for pages that satisfy the search criteria

Keywords × documents

All documents

All keywords	Document 1	Document 2	Document 3		Document *n*
“a”	1	1	1	…	1
“cat”	1	1	0	…	0
“database”	0	0	1	…	0
“dog”	0	1	0	…	1
“search”	0	0	1	…	0
	…	…	…	…	…

Signatures

- Store the matrix by columns
- For each document, store a signature
  - If the document satisfies a search condition (e.g., contains “database”), set the corresponding bit in the signature
  - Signature too big? Compress!
    - Example: hash keywords and then set corresponding bits
      - Lossy compression can generate false positives

hash(“database”) = 0110
hash(“dog”) = 1100
hash(“cat”) = 0010

doc$_1$ contains “database”: 0110
doc$_2$ contains “dog”: 1100
doc$_3$ contains “cat” and “dog”: 1110

Does *doc*$_3$ contain “database”?

Inverted lists versus signatures

- Inverted lists
  - High space overhead: could be bigger than the original documents!
- Signatures
  - Sequential scan through the signatures required