Web Searching & Indexing

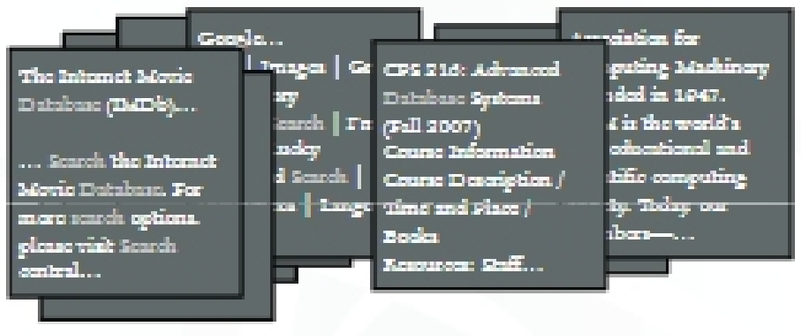
CPS 116

Introduction to Database Systems

Announcements (November 29)

- Homework #4 due today
- Course project demo Dec. 7-14
  - Each project gets a 30-minute slot with me and Yi
  - Watch for an email this weekend scheduling demo slots
- Final exam on Saturday, Dec. 15, 7-10pm
  - Again, open book, open notes
  - Focus on the second half of the course

Keyword search



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

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
- Signature files: store the matrix by columns

Inverted lists

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

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"
  -
- OR? NOT?
  -

What are "all" the keywords?

- All sequences of letters (up to a given length)?
  - … 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, e.g., a, of, the, it
  - Not useful in search
- Combine words with common stems
  - Example: database, databases
  - They can be treated as the same for the purpose of search

Frequency and proximity

- Frequency
  - $\langle keyword, \{ \quad \langle doc\text{-}id, number\text{-}of\text{-}occurrences \rangle, \langle doc\text{-}id, number\text{-}of\text{-}occurrences \rangle, \ldots \}\rangle$
- Proximity (and frequency)
  - $\langle keyword, \{ \quad \langle doc\text{-}id, \langle position\text{-}of\text{-}occurrence_1, position\text{-}of\text{-}occurrence_2, \ldots \rangle, \langle doc\text{-}id, \langle position\text{-}of\text{-}occurrence_1, \ldots \rangle \rangle, \ldots \}\rangle$
  - When doing AND, check for positions that are near

Signature files

- Store the matrix by columns and compress them
- For each document, store a w-bit signature
- Each word is hashed into a w-bit value, with only $s < w$ bits turned on
- Signature is computed by taking the bit-wise OR of the hash values of all words on the document

		Does doc_3 contain "database"?
$hash$("database") = 0110	doc_1 contains "database": 0110	
$hash$("dog") = 1100	doc_2 contains "dog": 1100	
$hash$("cat") = 0010	doc_3 contains "cat" and "dog": 1110	

- ☛ Some false positives; no false negatives