Index Construction Introduction to Information Retrieval **Donald J. Patterson**

Content adapted from Hinrich Schütze http://www.informationretrieval.org

Review

Connectivity Server

- Abstract Matrix
- Pairs of URLs (From -> To)
- Pairs of DocIDs
- Template based compression



Index Construction

Overview

- Introduction
- Hardware
- BSBI Block sort-based indexing
- SPIMI Single Pass in-memory indexing
- Distributed indexing
- Dynamic indexing
- Miscellaneous topics

The index has a list of vector space models



Letter from dead sister haunts brothers

Every time Julie Jensen's brothers hear the letter read, it brings everything back. Most of all, they wonder if they could have saved her. Her husband now stands trial for allegedly killing her. "I pray I'm wrong + nothing happens," Julie wrote days before her 1998 death. full story 1 1998 1 have 1 Everv 1 hear 1 Her 3 her 11 1 l'm 1 husband 1 Jensen's 1 if 2 Julie 1 it 1 killing 1 Letter 1 letter 1 Most 1 all 1 nothing allegedly 1 now 1 back 1 of 1 before 1 pray 1 brings 1 read. 2 brothers 1 saved 1 could 1 sister 1 stands 1 days 1 dead 1 story 1 death 1 the 1 everything 2 they 1 for 1 time 1 from 1 trial 1 full 1 wonder 1 happens 1 wrong haunts 1 wrote

"Term-Document Matrix" Capture Keywords

A Column for Each Web Page (or "Document")





- This picture is deceptive it is really very sparse
- Our queries are terms not documents
- We need to "invert" the vector space model
- To make "postings"

Terms

- Inverted index
 - (Term, Document) pairs
 - building blocks for working with Term-Document Matrices
- Index construction (or indexing)
 - The process of building an inverted index from a corpus
- Indexer
 - The system architecture and algorithm that constructs the index

Introduction The index is built from term-document pairs



Letter from dead sister haunts brothers

Every time Julie Jensen's brothers hear the letter read, it brings everything back. Most of all, they wonder if they could have saved her. Her husband now stands trial for allegedly killing her. "I pray I'm wrong + nothing happens," Julie wrote days before her 1998 death. full story

(TERM, DOCUMENT)

(1998,www.cnn.com) (Every,www.cnn.com) (Her,www.cnn.com) (I,www.cnn.com) (l'm,www.cnn.com) (Jensen's,www.cnn.com) (Julie,www.cnn.com) (Letter, www.cnn.com) (Most,www.cnn.com) (all,www.cnn.com) (allegedly,www.cnn.com) (back,www.cnn.com) (before,www.cnn.com) (brings,www.cnn.com) (brothers,www.cnn.com) (could,www.cnn.com) (days,www.cnn.com) (dead,www.cnn.com) (death,www.cnn.com) (everything,www.cnn.com) (for,www.cnn.com) (from,www.cnn.com) (full,www.cnn.com) (happens,www.cnn.com) (haunts,www.cnn.com)

(have,www.cnn.com) (hear,www.cnn.com) (her,www.cnn.com) (husband,www.cnn.com) (if.www.cnn.com) (it,www.cnn.com) (killing,www.cnn.com) (letter,www.cnn.com) (nothing,www.cnn.com) (now,www.cnn.com) (of,www.cnn.com) (pray,www.cnn.com) (read,,www.cnn.com) (saved,www.cnn.com) (sister,www.cnn.com) (stands,www.cnn.com) (story,www.cnn.com) (the,www.cnn.com) (they,www.cnn.com) (time,www.cnn.com) (trial,www.cnn.com) (wonder,www.cnn.com) (wrong,www.cnn.com) (wrote,www.cnn.com)

The index is built from term-document pairs

(TERM, DOCUMENT)

(1998,www.cnn.com) (Every,www.cnn.com) (Her,www.cnn.com) (I,www.cnn.com) (I'm,www.cnn.com) (Jensen's,www.cnn.com) (Julie,www.cnn.com) (Letter,www.cnn.com) (Most,www.cnn.com) (all,www.cnn.com) (allegedly,www.cnn.com) (back,www.cnn.com) (before,www.cnn.com) (brings,www.cnn.com) (brothers,www.cnn.com) (could,www.cnn.com) (days,www.cnn.com) (dead,www.cnn.com) (death,www.cnn.com) (everything,www.cnn.com) (for,www.cnn.com) (from,www.cnn.com) (full,www.cnn.com) (happens,www.cnn.com) (haunts,www.cnn.com)

(have,www.cnn.com) (hear,www.cnn.com) (her,www.cnn.com) (husband,www.cnn.com) (if,www.cnn.com) (it,www.cnn.com) (killing,www.cnn.com) (letter,www.cnn.com) (nothing,www.cnn.com) (now,www.cnn.com) (of,www.cnn.com) (pray,www.cnn.com) (read,,www.cnn.com) (saved,www.cnn.com) (sister,www.cnn.com) (stands,www.cnn.com) (story,www.cnn.com) (the,www.cnn.com) (they,www.cnn.com) (time,www.cnn.com) (trial,www.cnn.com) (wonder,www.cnn.com) (wrong,www.cnn.com) (wrote,www.cnn.com)

 Core indexing step is to sort by terms

Term-document pairs make lists of postings

(TERM, DOCUMENT, DOCUMENT, DOCUMENT,)

(1998,www.cnn.com,news.google.com,news.bbc.co.uk)
(Every,www.cnn.com, news.bbc.co.uk)
(Her,www.cnn.com,news.google.com)
(I,www.cnn.com,www.weather.com,)
(I'm,www.cnn.com,www.wallstreetjournal.com)
(Jensen's,www.cnn.com)
(Julie,www.cnn.com)
(Letter,www.cnn.com)
(all,www.cnn.com)
(allegedly,www.cnn.com)

- A posting is a list of all documents in which a term occurs.
- This is "inverted" from how documents
 naturally occur

Terms

• How do we construct an index?

Interactions

- An indexer needs raw text
 - We need crawlers to get the documents
 - We need APIs to get the documents from data stores
 - We need parsers (HTML, PDF, PowerPoint, etc.) to convert the documents
- Indexing the web means this has to be done web-scale

Construction

- Index construction in main memory is simple and fast.
- But:
 - As we build the index we parse docs one at a time
 - Final postings for a term are incomplete until the end.
 - At 10-12 postings per term, large collections demand a lot of space
 - Intermediate results must be stored on disk

Index Construction

Overview

- Introduction
- Hardware
- BSBI Block sort-based indexing
- SPIMI Single Pass in-memory indexing
- Distributed indexing
- Dynamic indexing
- Miscellaneous topics

System Parameters

- Disk seek time = 0.005 sec (2014: 0.004 hp 0.015 mobile)
- Transfer time per byte = 0.00000002 sec
- Processor clock rate = 0.00000001 sec
- Size of main memory = several GB
- Size of disk space = several TB

http://en.wikipedia.org/wiki/Hard disk drive performance characteristics

- Data is transferred from disk in blocks
 - Operating Systems read data in blocks, so
 - Reading one byte and reading one block take the same amount of time

- Disk Seek Time
 - The amount of time to get the disk head to the data
 - About 10 times slower than memory access
 - We must utilize caching
 - No data is transferred during seek
- Data is transferred from disk in blocks
 - There is no additional overhead to read in an entire block
 - How long would it take to get 10 MB if it is one block?
 - How long would it take to get 10 MB if it is stored in 100 blocks?

- Disk Seek Time
 - The amount of time to get the disk head to the data
 - About 10 times slower than memory access
 - We must utilize caching
 - No data is transferred during seek
- Data is transferred from disk in blocks
 - There is no additional overhead to read in an entire block
 - 0.2098 seconds to get 10 MB if it is one block
 - 0.7048 seconds to get 10 MB if it is stored in 100 blocks

Solving data recovery time



System Parameters

• What about compressed disks?

- Data transfers are done on the system bus, not by the processor
 - The processor is not used during disk I/O
 - Assuming an efficient decompression algorithm
 - The total time of reading and then decompressing compressed data is usually less than reading uncompressed data.

Compressed Data on Disk



System Parameters

• What about SSDs?

Index Construction

Overview

- Introduction
- Hardware
- BSBI Block sort-based indexing
- SPIMI Single Pass in-memory indexing
- Distributed indexing
- Dynamic indexing
- Miscellaneous topics

BSB

Reuters collection example (approximate #'s)

- 800,000 documents from the Reuters news feed
- 200 terms per document
- 400,000 unique terms
- number of postings 100,000,000



Reuters collection example (approximate #'s)

- Sorting 100,000,000 records on disk is too slow because of disk seek time.
 - Parse and build posting entries one at a time
 - Sort posting entries by term
 - Then by document in each term
 - Doing this with random disk seeks is too slow
 - e.g. If every comparison takes 2 disk seeks and N items need to be sorted with N log2(N) comparisons?
 - How long is that going to take?

BSBI

Reuters collection example (approximate #'s)

- 100,000,000 records
- Nlog2(N) is = 2,657,542,475.91 comparisons
- 2 disk seeks per comparison = 13,287,712.38 seconds x 2
- = 26,575,424.76 seconds
- = 442,923.75 minutes
- = 7,382.06 hours
- = 307.59 days
- = 84% of a year
- = 1% of your life



Prof. Patterson waiting for your slow algorithm to end

Index Construction

Review

- termID is an index given to a vocabulary word
 - e.g., "house" = 57820
- docID is an index given to a document
 - e.g., "news.bbc.co.uk" = 74291
- posting list is a data structure for the term-document matrix



posting list is an inverted data structure

Different way to sort index

- 12-byte records (term, doc, meta-data)
- Need to sort T= 100,000,000 such 12-byte records by term
- Define a block to have 1,600,000 such records
 - can easily fit a couple blocks in memory
 - we will be working with 64 such blocks
- Accumulate postings for each block (real blocks are bigger)
- Sort each block
- Write to disk
- Then merge



Different way to sort index



BLOCKSORTBASEDINDEXCONSTRUCTION() 1 $n \leftarrow 0$

- 2 while (all documents not processed)
- 3 **do** $block \leftarrow PARSENEXTBLOCK()$
- 4 BSBI-INVERT(block)
- 5 WRITEBLOCKTODISK $(block, f_n)$
- 6 MERGEBLOCKS $(f_1, f_2..., f_n, f_{merged})$

Block merge indexing

- Parse documents into (TermID, DocID) pairs until "block" is full
- Invert the block
 - Sort the (TermID,DocID) pairs
 - Compile into TermID posting lists
- Write the block to disk
- Then merge all blocks into one large postings file
 - Need 2 copies of the data on disk (input then output)

Analysis of BSBI

- The dominant term is O(TlogT)
 - T is the number of (TermID,DocID) pairs
- But in practice ParseNextBlock takes the most time
- Then MergingBlocks
- Again, disk seeks times versus memory access times

Analysis of BSBI

- 12-byte records (term, doc, meta-data)
- Need to sort T= 100,000,000 such 12-byte records by term
- Define a block to have 1,600,000 such records
 - can easily fit a couple blocks in memory
 - we will be working with 64 such blocks
- 64 blocks * 1,600,000 records * 12 bytes = 1,228,800,000 bytes
- Nlog2N comparisons is 5,584,577,250.93
- 2 touches per comparison at memory speeds (10e-6 sec) =
 - 55,845.77 seconds = 930.76 min = 15.5 hours

WESTMONT COMPUTER SCIENCE

