# 605.744 Information Retrieval

Course Overview & Boolean Retrieval

JOHNS HOPKINS
WHITING SCHOOL
*of* ENGINEERING

# Outline

- **Course Pragmatics**
  - **Schedule of topics**
  - **Grading policy**
- Overview of Text Retrieval
- Boolean Model
  - Queries
  - Document Representations
- Tokenization

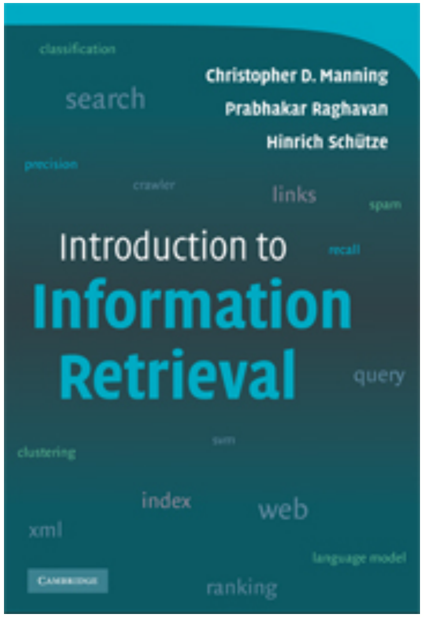
# Course Overview

- Basic theoretical understanding of IR
  - Representing and indexing text documents
  - Retrieval models
  - Implementing querying efficiently
- Application areas and research topics such as:
  - Text Classification
  - Cross-language retrieval
  - Retrieval on the Web
  - Speech Retrieval
- Assess IR performance:
  - Recall/Precision and other metrics
- Gain hands-on experience building an IR system
- Introduce related topics in computational linguistics

# Course Philosophy

- Approach combines lectures, seminar format, and laboratory assignments
  - Assigned readings and lectures are a primary tool for learning
  - Discussions & questions improve understanding
  - Homeworks and programming projects provide hands-on experience
  - Students present paper summaries and / or projects to whole class

# Course Text



Introduction to Information Retrieval

- Manning, Raghavan, and Schutze (2008)
  - http://nlp.stanford.edu/IR-book/information-retrieval-book.html

Other books:

- IR: Implementing and Evaluating Search Engines (2010)
  - Buettcher, Clarke, and Cormack
- Managing Gigabytes, 2nd edition (1999)
  - Witten, Moffat, & Bell
- IR: Algorithms and Heuristics (2004)
  - Grossman and Frieder
- Modern Information Retrieval 2nd ed. (2011)
  - Baeza-Yates and Ribeiro-Neto

# Programming Assignments

- Practical labs
  - Writing programs to work with text datasets

- Five programming assignments
  - 30% of grade
  - A couple of weeks apart
  - First three build up to crating a small, but true to practice text retrieval engine

# Short Problem Sets

- Questions based on readings and lecture content

- Almost weekly
  - 20% of grade
  - A couple of weeks apart
  - First three build up to crating a small, but true to practice text retrieval engine

- Problem Sets are assessed with the following rubric
  - Correctness: 70%
  - Clarity: 10%
  - Justification: 10%
  - Completeness: 10%

# Exam

- One open-book exam
  - 15% of grade

- Tests knowledge of course concepts

- Emphasizes calculations, short problems, analysis

# Scholarly Engagement

- Online Discussions
  - 10% of grade
  - Not merely responding to instructor-posed questions
  - Really prefer to be student driven

- Assessed four times during semester
  - Demonstrates knowledge of content
  - Critical thinking
  - Frequency
  - Clarity
  - Value to others / promotes learning

# Research Paper Summary

- Review of academic paper
  - 5% of grade
- Written review is shared with the class
- Reviewer also responds to questions posed about the paper in the discussion forum

# Class Project

- Goal is to research an area or develop an idea that you would like to explore in greater depth
  - 20% of grade
  - Optional: highest course grade is B+ without submitting a project

- Deliverables include a written report (5-8 pages) and a short video presentation

- More details in 3-4 weeks

# Project Ideas

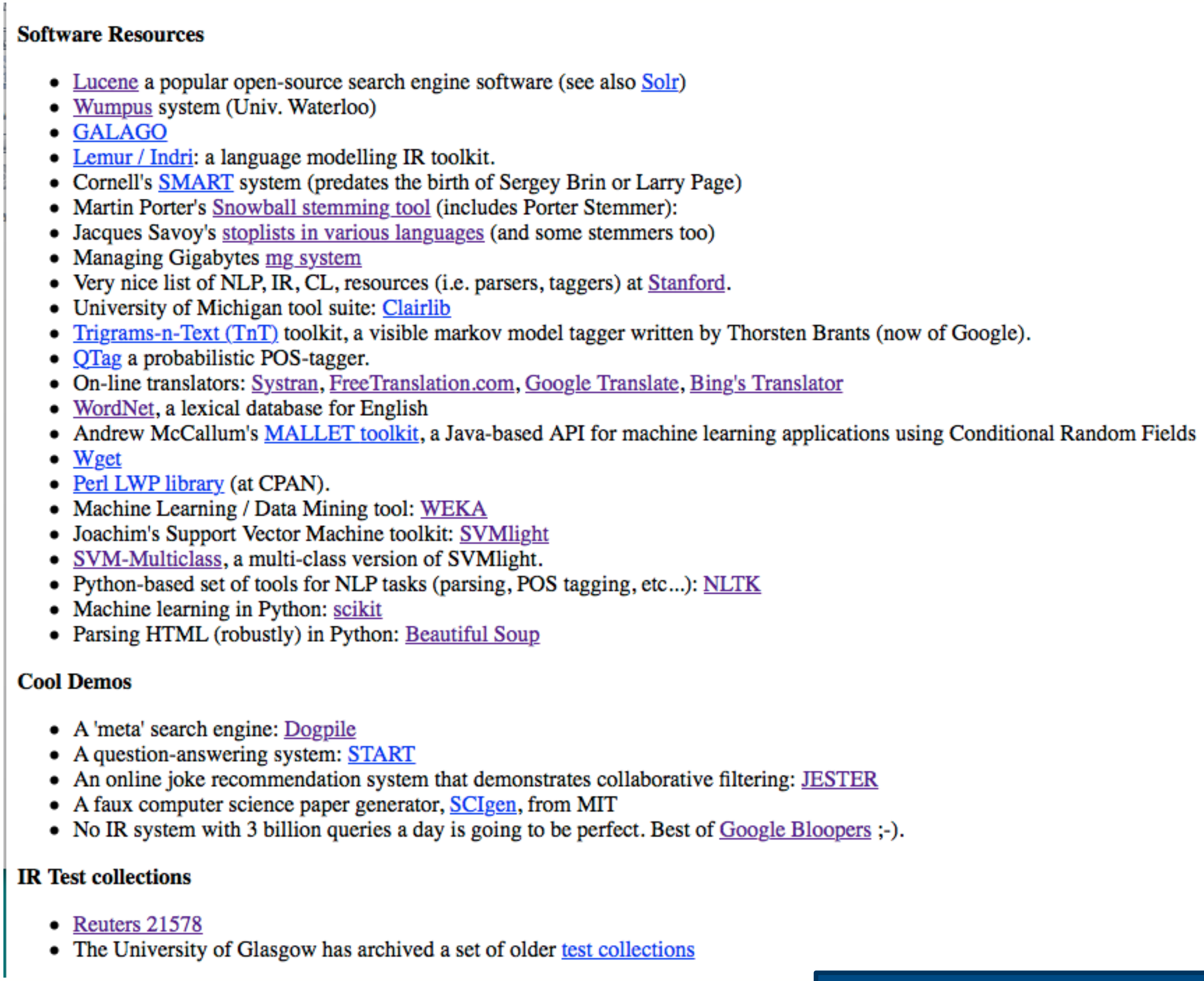
- Analyzing police crime reports and classifying narratives by type of criminal activity
- Exploring methods to compress indexes using document-identifier reassignment
- Extraction of fields from Craigslist apartment rentals (i.e., automatic identification of the number of rooms, monthly rent, location, if smoking is allowed, etc…)
- Predicting attributes of document authorship (e.g., author gender, century of authorship, or who authored a particular document)
- Predicting stock price movement using open source financial data (e.g., Twitter streams, SEC filings)

# Course Grades (Summary)

| | |
|---|---|
| Programming Assignments | 30% |
| Short Problem Sets | 20% |
| Final Exam | 15% |
| Scholarly Engagement | 10% |
| Paper Summary | 5% |
| Class Project | 20% (required for A- or better) |

Please carefully read the course syllabus

# Resources



### Software Resources

- Lucene a popular open-source search engine software (see also Solr)
- Wumpus system (Univ. Waterloo)
- GALAGO
- Lemur / Indri: a language modelling IR toolkit.
- Cornell's SMART system (predates the birth of Sergey Brin or Larry Page)
- Martin Porter's Snowball stemming tool (includes Porter Stemmer):
- Jacques Savoy's stoplists in various languages (and some stemmers too)
- Managing Gigabytes mg system
- Very nice list of NLP, IR, CL, resources (i.e. parsers, taggers) at Stanford.
- University of Michigan tool suite: Clairlib
- Trigrams-n-Text (TnT) toolkit, a visible markov model tagger written by Thorsten Brants (now of Google).
- QTag a probabilistic POS-tagger.
- On-line translators: Systran, FreeTranslation.com, Google Translate, Bing's Translator
- WordNet, a lexical database for English
- Andrew McCallum's MALLET toolkit, a Java-based API for machine learning applications using Conditional Random Fields
- Wget
- Perl LWP library (at CPAN).
- Machine Learning / Data Mining tool: WEKA
- Joachim's Support Vector Machine toolkit: SVMlight
- SVM-Multiclass, a multi-class version of SVMlight.
- Python-based set of tools for NLP tasks (parsing, POS tagging, etc...): NLTK
- Machine learning in Python: scikit
- Parsing HTML (robustly) in Python: Beautiful Soup

### Cool Demos

- A 'meta' search engine: Dogpile
- A question-answering system: START
- An online joke recommendation system that demonstrates collaborative filtering: JESTER
- A faux computer science paper generator, SCIgen, from MIT
- No IR system with 3 billion queries a day is going to be perfect. Best of Google Bloopers ;-).

### IR Test collections

- Reuters 21578
- The University of Glasgow has archived a set of older test collections

http://pmcnamee.net/ir.html

# Research Software Systems

- Wumpus
  - U. Waterloo (Open source, C++)
- Terrier
  - Glasgow (Open source, Java)
- Lemur / Indri / Galago
  - Carnegie Mellon / UMass (C++ & Java bindings)
- Lucene
  - Apache (Java)
- SMART
  - Developed at Cornell University (C)
- mg
  - From the authors of *Managing Gigabytes* (C)
- INQUERY
  - Univ. Massachusetts (Amherst). Available?

# Outline

- Course Pragmatics
  - Schedule of topics
  - Grading policy
- **Overview of Text Retrieval**
- Boolean Model
  - Queries
  - Document Representations
- Tokenization

# IR Overview

- What is Information Retrieval?
  - How does it differ from database querying?
- History of IR
  - Field over 40 years old
  - Why so popular now?
  - Impact of the Web

*I never waste memory on things that can easily be stored and retrieved from elsewhere – A. Einstein*

# What is Information Retrieval?

- Field concerned with the organization, storage, and retrieval of information
  - Especially text
  - Also retrieval of semistructured data (XML), video images, speech, music
- Requires algorithms and data structures
  - For manipulating natural language
  - To efficiently store and process data
- Related fields
  - Natural Language Processing, Library Science
  - Computational Linguistics, Digital Libraries

# What makes IR a hard problem

- Under good circumstances
  - Text is unstructured
  - In the hardest cases, it requires understanding of semantics
  - Human language presents distinct problems (e.g., ambiguity)
- Under hard circumstances
  - Patent retrieval: applications tend to use low content words; why?
  - One estimate is that 40% of web pages change monthly, many pages ‘lie’ about their content, new pages aren’t linked to
- Multimedia information
  - Hard to store (size), represent, and compare

# IR vs. DB: text is unstructured

- RDBS

  *SELECT SALARY FROM EMPTBL WHERE BASEPAY > 75000*

- *Text Retrieval*

  *"Find salary surveys for CS/IT professionals in the Washington DC area"*

- *SQL semantics are clearly specified*
  - A single omission results in a completely incorrect response to a query
  - Language is less well-defined; missing one relevant document might not be catastrophic

# Nuance in Language

- Find salary surveys for CS/IT professionals in:
  - Seattle, Washington
  - Washington, DC

- List professional sports teams in Baltimore, except the Orioles

- Fortune 500 CEOs who are not male

# Three Major Problems in IR

- Polysemy
  - Words can have multiple meanings
- Synonymy
  - The same concept can be expressed using different words
- Morphology
  - Many word forms are related

    *juggle, juggling, juggled, jugglers*

    *go, going, went*

  - Small affixes adjust meaning

# Polysemy

- Ambiguity pervasive
  - jaguar, bank, see, hornet, red, aa,
- Distinctions vary in granularity
  - cool (popular) vs. cool (low in temperature)
  - list (to name items in a list) vs. (to include in a list)

*Hornet: insect, NBA team, or fighter aircraft?*

*See: to percieve visually, or the Holy See (a name for the Vatican)*

# Synonymy

- English provides no canonical way to reference people and things
  - President Carter, Pres. Carter, Jimmy Carter; the 39th president, Rosalynn Carter's husband

- Speakers of a language learn preferential ways of expressing things:
  - strong tea / powerful computers

- Documents have a limited vocabulary with discrete occurrences; words have many synonyms
  - query: 'fast automobiles'

    *should match 'fast cars', 'speedy cars', …*

# Morphology

- In language there isn’t a 1-to-1 mapping between words and concepts

- We’d like a query using the word ‘airplane’ to match documents that may not contain airplane, but contain related word forms
  - airplane: airplanes, aircraft, planes
  - actor: actor or actress

- Will revisit the topic later

# Pre-history of IR

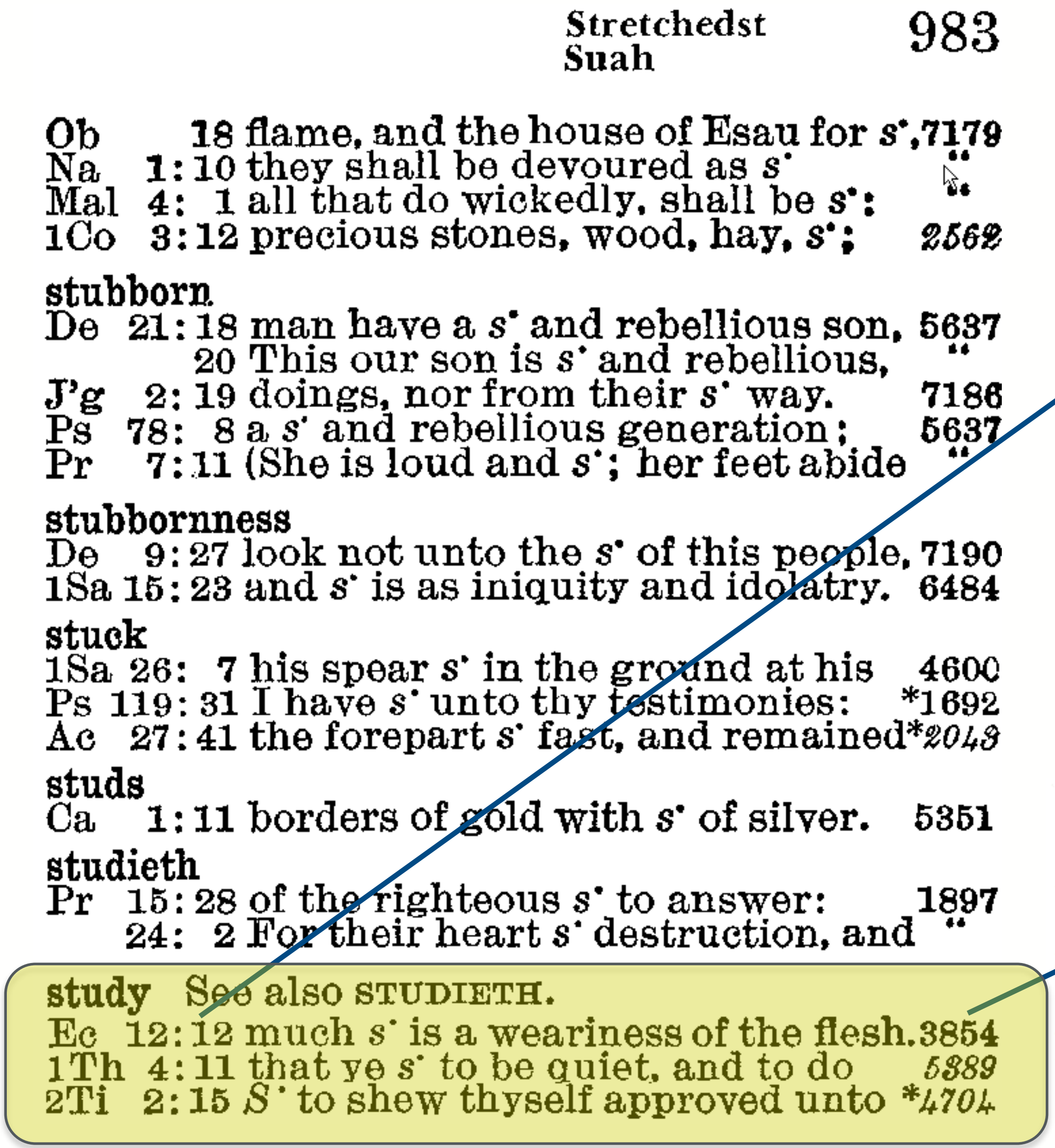
- 300 BCE Euclid’s treatise,The Elements
- 300 BCE Ptolemy I founds Great Library at Alexandria which grows to include 700,000+ volumes (scrolls)
- 391 Great Library destroyed by fanatics (implications for the Web?)
- 600 Number 0 used in India
- 825 Muhammad ibn Musa Al-Khowarizmi writes treatise on algebra; the English word algorithm is derived from his name
- 1230s St. Anthony (of Padova) creates concordance for Latin Vulgate
- 1247 Cardinal Hugo employs 500 monks to build a concordance
- 1470s Johannes Gutenberg creates printing press
- 1550 First English concordance of entire Bible
- 1640 Blaise Pascal develops mechanical calculator. It performed subtraction by adding complements
- 1714 Henry Mills conceives of the typewriter
- 1837 Morse Code is an early text encoding scheme
- 1857 Sir Charles Wheatstone stores Morse codes on paper tapes; they could be prepared offline and transmitted later

# The Great Library Rebuilt (2002)

# Industrial Age Computing

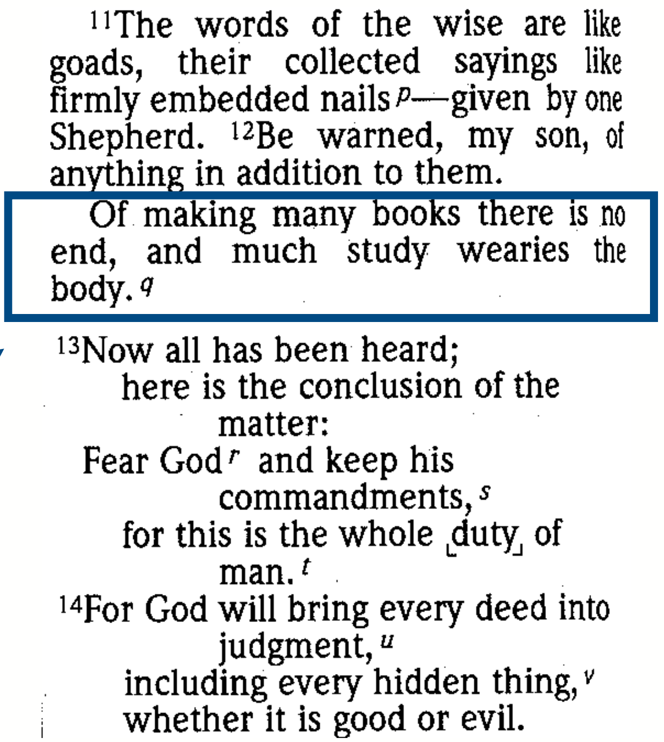
- 1867 First commercial typewriter available
- 1872 21-year old Melvil Dewey invents a classification code
- 1890 Hollerith's punched cards used to tabulate census information automatically (Hollerith's company CTR later became IBM)
- 1890 Dr. James Strong (and students) create an 'exhaustive' concordance
- 1900 John Ambrose invents the vacuum tube
- 1936 Konrad Zuse applies for patent for programmable memory
- 1937 Alan Turning invents the Turing Machine
- 1941 Harvard Mark I computer (Howard Aiken and Thomas J. Watson Sr.)
- 1943 ENIAC construction begins
- 1945 Vannever Bush conceives of MEMEX device ("As we may think" in Atlantic Monthly)
- 1946 ENIAC unveiled
- 1947 Point-contact transistor developed at Bell Labs
- 1948 Claude Shannon's work in information theory, coins term 'bit'

# Entry from Strong's Concordance



Stretchedst
Suah 983

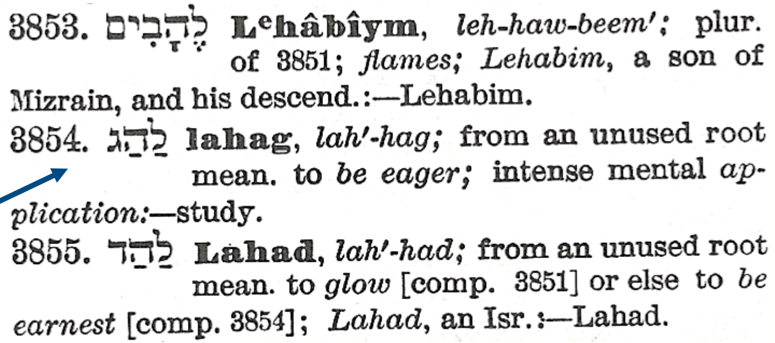
| | | | |
|---|---|---|---|
| Ob | 18 | flame, and the house of Esau for *s*˙, | 7179 |
| Na | 1:10 | they shall be devoured as *s*˙ | " |
| Mal | 4: 1 | all that do wickedly, shall be *s*˙: | " |
| 1Co | 3:12 | precious stones, wood, hay, *s*˙; | *2562* |
| **stubborn** | | | |
| De | 21:18 | man have a *s*˙ and rebellious son, | 5637 |
| | 20 | This our son is *s*˙ and rebellious, | " |
| J'g | 2:19 | doings, nor from their *s*˙ way. | 7186 |
| Ps | 78: 8 | a *s*˙ and rebellious generation; | 5637 |
| Pr | 7:11 | (She is loud and *s*˙; her feet abide | " |
| **stubbornness** | | | |
| De | 9:27 | look not unto the *s*˙ of this people, | 7190 |
| 1Sa | 15:23 | and *s*˙ is as iniquity and idolatry. | 6484 |
| **stuck** | | | |
| 1Sa | 26: 7 | his spear *s*˙ in the ground at his | 4600 |
| Ps | 119:31 | I have *s*˙ unto thy testimonies: | *1692 |
| Ac | 27:41 | the forepart *s*˙ fast, and remained | **2043* |
| **studs** | | | |
| Ca | 1:11 | borders of gold with *s*˙ of silver. | 5351 |
| **studieth** | | | |
| Pr | 15:28 | of the righteous *s*˙ to answer: | 1897 |
| | 24: 2 | For their heart *s*˙ destruction, and | " |
| **study** | | See also STUDIETH. | |
| Ec | 12:12 | much *s*˙ is a weariness of the flesh. | 3854 |
| 1Th | 4:11 | that ye *s*˙ to be quiet, and to do | *5389* |
| 2Ti | 2:15 | *S*˙ to shew thyself approved unto | **4704* |



[11]The words of the wise are like goads, their collected sayings like firmly embedded nails[p]—given by one Shepherd. [12]Be warned, my son, of anything in addition to them.

Of making many books there is no end, and much study wearies the body.[q]

[13]Now all has been heard;
here is the conclusion of the matter:
Fear God[r] and keep his commandments,[s]
for this is the whole ⌞duty⌟ of man.[t]
[14]For God will bring every deed into judgment,[u]
including every hidden thing,[v]
whether it is good or evil.



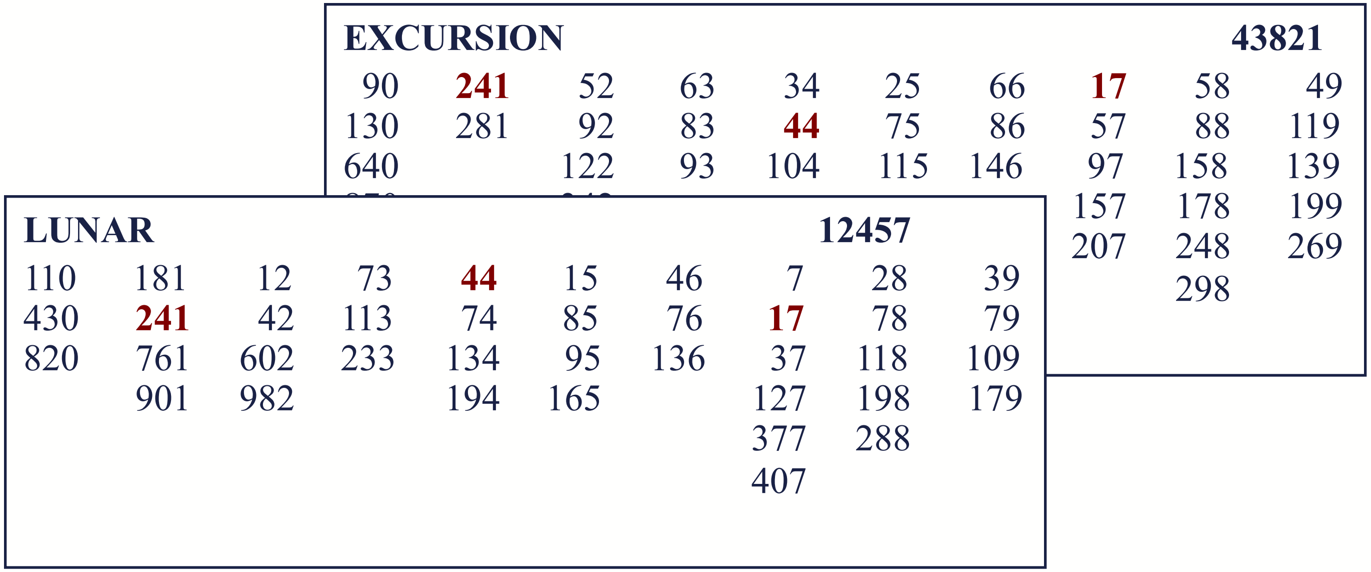
3853. לְהָבִים **Lᵉhâbîym**, *leh-haw-beem'*; plur. of 3851; *flames; Lehabim*, a son of Mizrain, and his descend.:—Lehabim.

3854. לַהַג **lahag**, *lah'-hag*; from an unused root mean. to *be eager*; intense mental *application*:—study.

3855. לַהַד **Lahad**, *lah'-had*; from an unused root mean. to *glow* [comp. 3851] or else to *be earnest* [comp. 3854]; *Lahad*, an Isr.:—Lahad.

# Early (Manual) IR Systems

- Mortimer Taube
  - Punched cards on IBM hardware
- Uniterm (Casey, Perry, Berry, Kent: 1958 –developed and used from mid 1940’s)



**EXCURSION** **43821**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 90 | **241** | 52 | 63 | 34 | 25 | 66 | **17** | 58 | 49 |
| 130 | 281 | 92 | 83 | **44** | 75 | 86 | 57 | 88 | 119 |
| 640 | | 122 | 93 | 104 | 115 | 146 | 97 | 158 | 139 |
| | | | | | | | 157 | 178 | 199 |
| | | | | | | | 207 | 248 | 269 |
| | | | | | | | | 298 | |

**LUNAR** **12457**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 110 | 181 | 12 | 73 | **44** | 15 | 46 | 7 | 28 | 39 |
| 430 | **241** | 42 | 113 | 74 | 85 | 76 | **17** | 78 | 79 |
| 820 | 761 | 602 | 233 | 134 | 95 | 136 | 37 | 118 | 109 |
| | 901 | 982 | | 194 | 165 | | 127 | 198 | 179 |
| | | | | | | | 377 | 288 | |
| | | | | | | | 407 | | |

# Advent of Computer Science

- 1962 First Comp Sci. degree program offered by Purdue U.
- 1963 ASCII standard developed
- 1965 CD-ROM technology invented (James Russell)
- 1969 ARPANET contains 4 hosts (23 in 1971)
- 1969 UNIX operating system (Ritchie & Thompson)
- 1972 Tomlinson sends first email message
- 1975 Microsoft founded by Gates and Allen
- 1977 Apple II personal computer
- 1981 IBM PC
- 1982 TCP/IP basis for NSFNet
- 1984 Apple Macintosh with windowing interface
- 1984 1,000 Internet hosts
- 1988 Robert Morris, a Cornell U. graduate student, unleashes the ‘Internet Worm’
- 1989 100,000 Internet hosts

# Birth of the Web

- 1989 Tim Berners-Lee invents World-Wide-Web
- 1992 1,000,000 Internet hosts, but only 50 web sites
- 1994 Two Stanford graduate students found Yahoo, a manually build on-line directory
- 1995 AltaVista indexes 15 million web pages
- 1996 Two other Stanford graduate students collaborate on Google
- 1997 Lawrence and Giles paper characterizing Web
- 1999 Excite search engine sold for $6.7 billion; around same time automotive division of Volvo sold for $6.3 billion.
- 2000 1 billion web pages on public web; 10 million web sites, 93 million or so Internet hosts
- 2002 Google claims 3 billion page index
- 2004 Google IPO
- 2006 Google's stock value exceeds $150 billion (> Coke, IBM, AT&T)
- 2009 Microsoft rebrands Web search as Bing

Sources:

http://www.mcs.net/~jorn/html/net/timeline.html

http://www.maxmon.com/history.htm

http://www.let.leidenuniv.nl/history/

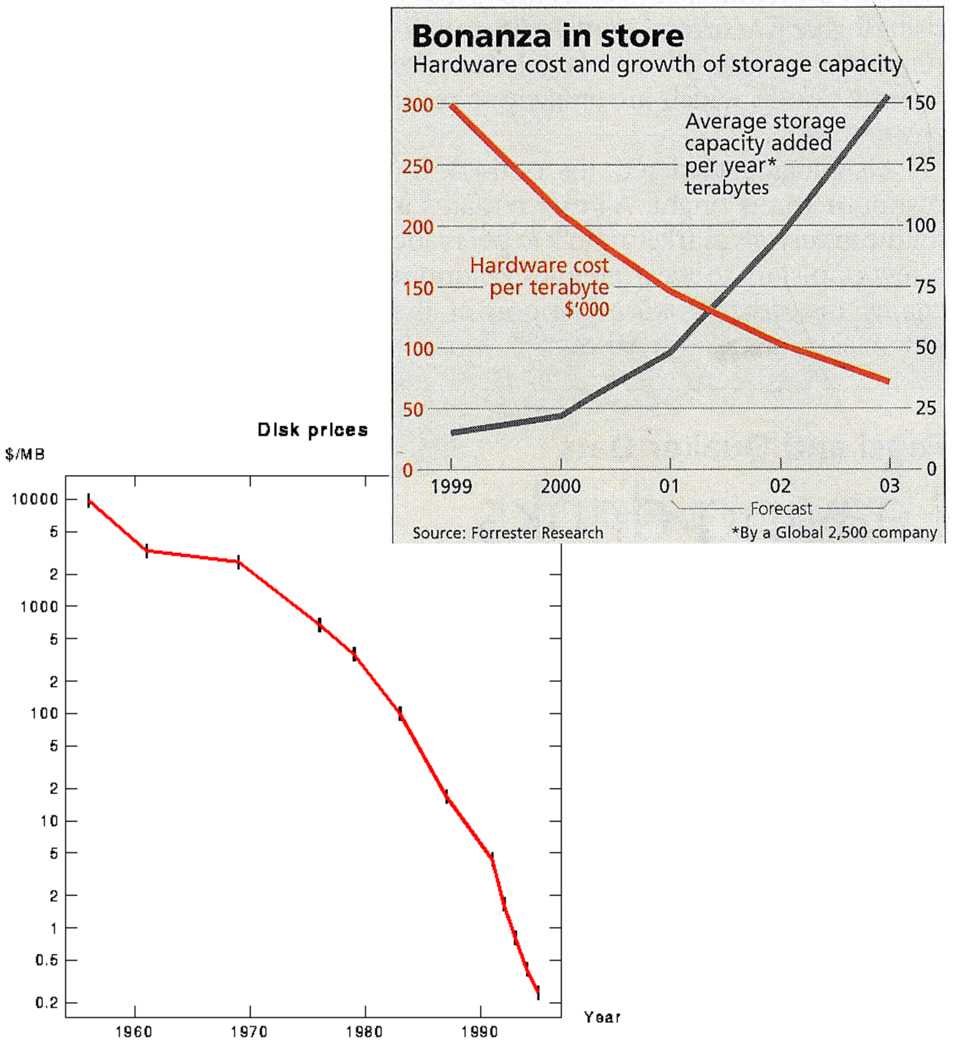
http://ei.cs.vt.edu/~history/

http://www.computerhistory.org/

# Why is IR thriving today?

- Dropping prices for external storage is a significant factor

- Other factors
  - Increased expectations and demonstrated utility
  - Web 2.0 / computational advertising
  - Ease of use



From www.lesk.com

# Changing Perspectives

- 40 years ago
  - Classification and categorization (automating librarians)
  - Systems and query languages
  - Niche domains (chemistry, legal, medical)
  - Focus on keywords or abstract search of library records

- Advent of the Web
  - Free (low cost) universal access
  - No central editorial board
  - Search becomes something ordinary users can do

# Trends (last 15 years)

- Not only the Web, but also corporate intranets
- Multimedia retrieval
- Users don't really want ranked lists of documents
    - Informative, Navigational, & Transactional queries
    - Question-Answering
- Semi-structured data: XML, RDF
- Information is increasingly multilingual
- Personalization / Digital Privacy
- Sophistication in generating revenue
- Facebook: social search?

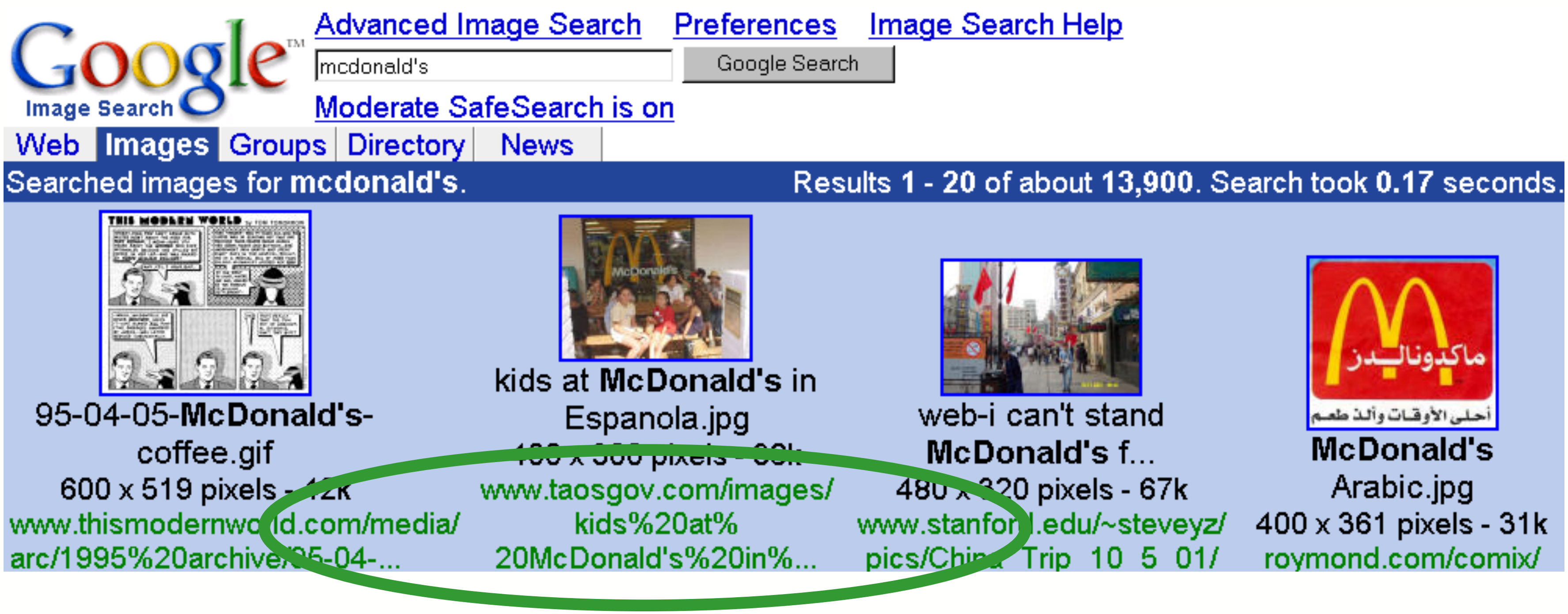
# Enterprise Search

- Every large company now wants to search their own internally produced content
  - Without exposing it to the Internet
- For about 10 years Google has sold “Google appliances”
  - Plug a box into your network, and create a local Google instance.
- Apache Lucence, Solr, and Elastic Search make it feasible for sysadmins to manage internal search

# Beyond Text

- Images
  - Content based methods are difficult
  - Can try to make inferences based on filenames or coordinate text
  - Take up much more storage than text
- Video
  - Usually use sampled sequences of images
- Broadcast speech
  - 1000s of radio stations from around the world
  - Typical approach: transcribe speech into text (with errors) and treat as ‘normal’ text
- Scanned text
  - Like speech, scan (w/ errors) and index
- Maps, Diagrams, Music (open problems)

# Google Images



Advanced Image Search    Preferences    Image Search Help

mcdonald's    Google Search

Moderate SafeSearch is on

Searched images for **mcdonald's**.    Results **1** - **20** of about **13,900**. Search took **0.17** seconds.

95-04-05-**McDonald's**-coffee.gif
600 x 519 pixels - 42k
www.thismodernworld.com/media/arc/1995%20archive/95-04-...

kids at **McDonald's** in Espanola.jpg
400 x 300 pixels - 60k
www.taosgov.com/images/kids%20at%20McDonald's%20in%...

web-i can't stand **McDonald's** f...
480 x 320 pixels - 67k
www.stanford.edu/~steveyz/pics/China_Trip_10_5_01/

**McDonald's** Arabic.jpg
400 x 361 pixels - 31k
roymond.com/comix/

document text basis for search

# Beyond Document Retrieval

- User’s typically do not want to merely find documents of interest

- A. Broder (CTO AltaVista) taxonomy (11/00)
  - Informational needs
  - Navigation (e.g., surrogate bookmarks)
  - Transactional

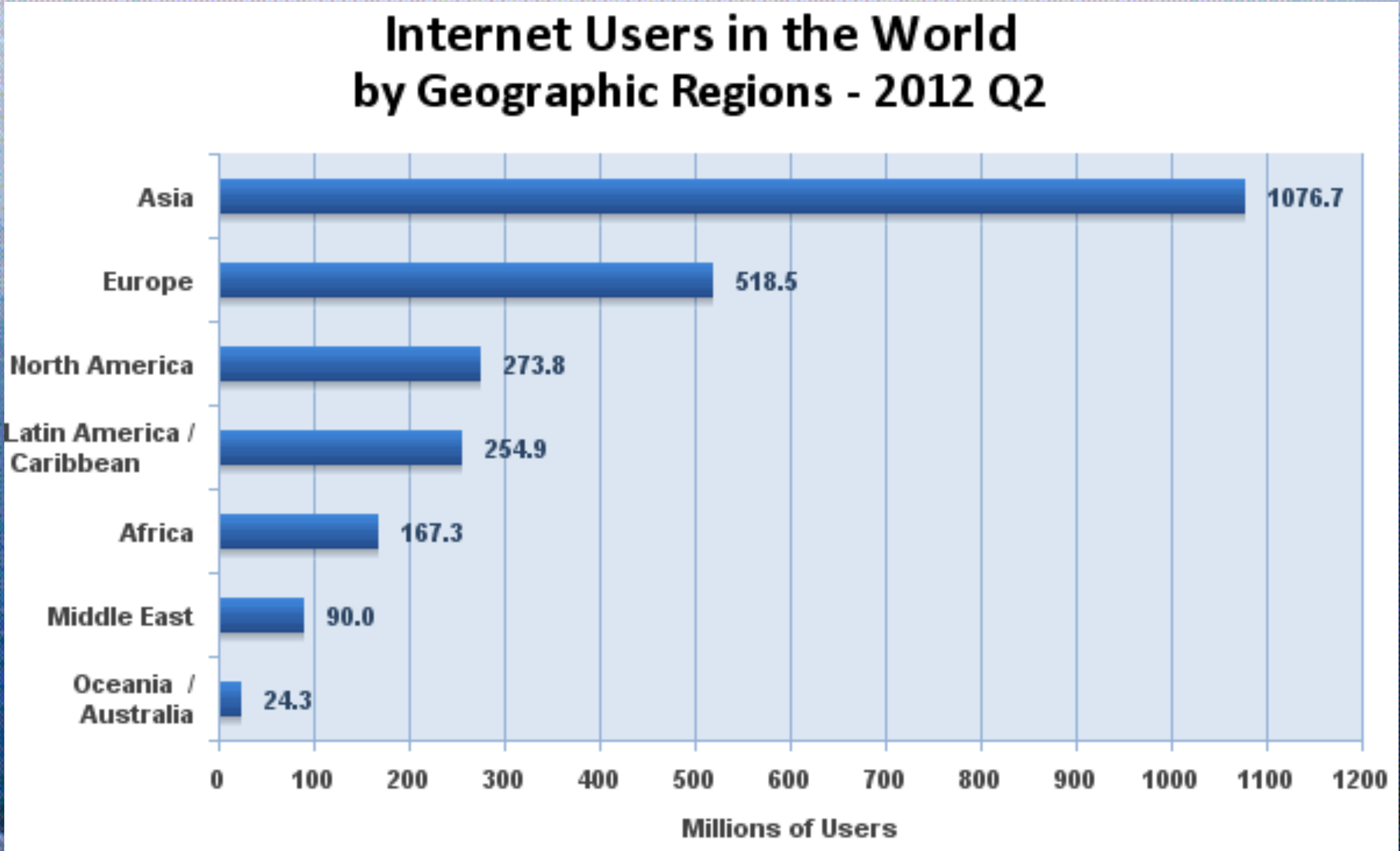
- Question Answering
  - J. Prange (IARPA), Advanced Question-Answering
  - Yahoo Answers, eHow

# Question-Answering Systems

- FAQ-Finder
  - Indexes FAQ lists and tries to find responsive answers to common questions
- Yahoo Answers
  - Looks for web pages likely to contain answers to common, simple questions

    *(e.g., “How do I make an apple pie?”)*
- eHow
  - Web 2.0: free, user-contributed, ranked answers to common questions
- Text Retrieval Community studied QA for several years
  - TREC-8 evaluation (1999) was the first

# Beyond English



Internet Users in the World
by Geographic Regions - 2012 Q2

Source: Internet World Stats - www.internetworldstats.com/stats.htm
2,405,518,376 Internet users estimated for June 30, 2012
Copyright © 2012, Miniwatts Marketing Group

# Beyond Single Requests

- 1999: Infoseek anecdotally reports
  - ~50M queries / day
  - ~600 queries / second over 10^8 collection

- 2010 estimates
  - Google: 1-3 billion / day
  - Yahoo: 180 M / day
  - Bing: 80 M / day

- Ideally, user context should be leveraged
  - System can learn a profile over time
  - Benefits successive queries

# Beyond Surfing: Text Classification

- Dual problem to ad hoc retrieval
  - Filter incoming messages relevant to a defined profile

    *Push technology vs. pull*

    *Examples: Bloomberg news, Book or movie recommendations, targeted advertisements, spam filtering*

- Scenario:
  - You are a safety engineer for a large automotive manufacturer. You want to keep track of reports of accidents in a new vehicle

    *Don't have access to a static collection of documents; instead, news stories and reports trickle in over time; relevance decisions must be made immediately*

    *Can't be plagued by too many false alarms, but also don't want to miss relevant reports*

# 605.744 – Information Retrieval

Boolean Model of Retrieval

*The Feynman Problem-Solving Algorithm: (1) write down the problem; (2) think very hard; (3) write down the answer. – Murray Gellmann*

JOHNS HOPKINS
WHITING SCHOOL
*of* ENGINEERING

# Outline

- Course Pragmatics
  - Schedule of topics
  - Grading policy
- Overview of Text Retrieval
- **Boolean Models**
  - Queries
  - Document Representations
- Tokenization

# Boolean Model

- Documents are sets of terms (read word)
- Likewise, queries are sets of terms
- The framework is set-theory
  - Based on work of George Boole (1850)
- Relevant documents are determined using set operations (set-membership)
  - Ex: query = “rabies AND shot”
  - Any document containing both terms is considered relevant
- Standard operations: AND, OR, NOT

# Boolean Queries

- INFIX operators
  - ((cat AND dog) OR (collar AND leash))
- NOT is UNARY PREFIX operator
  - ((cat AND dog) OR (collar AND (NOT dog)))
- AND and OR are n-ary operators
  - (cat AND dog AND rabies AND shot)
- De Morgan's Laws
  - NOT(a) AND NOT(b) = NOT(a OR b)
  - NOT(a) OR NOT(b)= NOT(a AND b)
  - NOT(NOT(a)) = a

# Boolean Queries

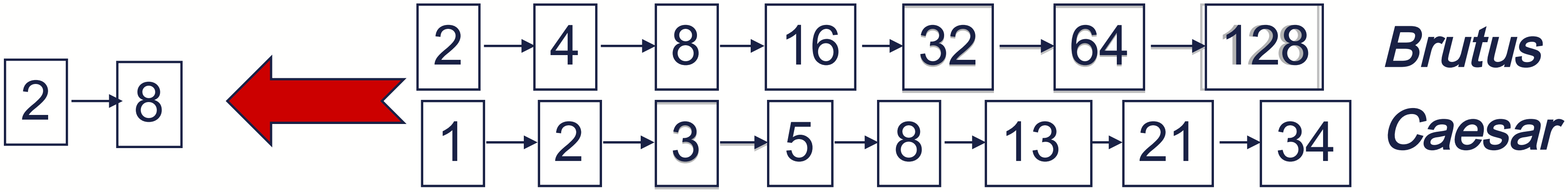
- (Cat OR Dog) AND (Collar OR Leash)
  - Which of the following combinations satisfies this statement:

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|
| Cat | | x | | x | | x | x |
| Dog | | x | | x | x | x | x |
| Collar | x | | | x | x | | x |
| Lease | | x | x | | | | x |

- (Cat OR Dog): D2, D4, D5, D6, D7
- (Collar OR Leash): D1, D2, D3, D4, D5, D7
- ANDing: D2, D4, D5, & D7

# The merge (Boolean AND)

- Walk through the two postings simultaneously, in time linear in the total number of postings entries



If the list lengths are *x* and *y*, the merge takes O(*x*+*y*) operations.

Crucial: postings sorted by docID.

Courtesy of Manning and Raghavan

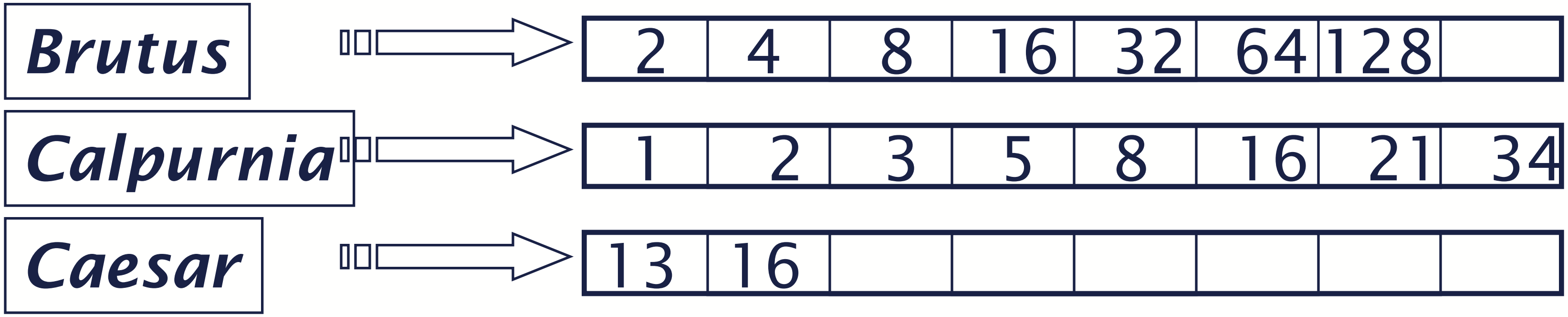
# Processing Boolean Queries

- If sorted document lists are available
  - A new ‘array’ can be created from existing arrays of documents
- Otherwise
  - Use a linear-time algorithm

    *Hashtables support union, intersection and set-difference*

# Query optimization

- What is the best order for query processing?
- Consider a query that is an *AND* of *t* terms.
- For each of the *t* terms, get its postings, then *AND* them together.



**Brutus** → 2 4 8 16 32 64 128

**Calpurnia** → 1 2 3 5 8 16 21 34

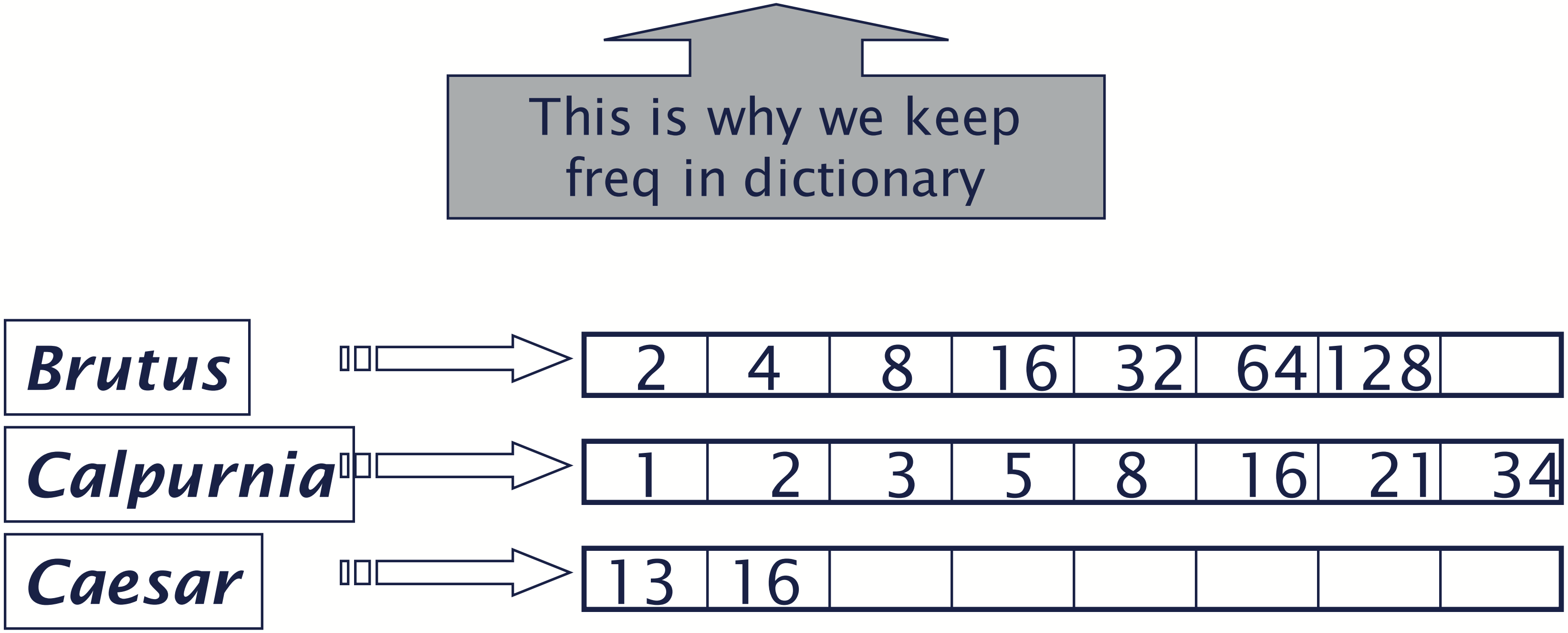
**Caesar** → 13 16

Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

Courtesy of Manning and Raghavan

# Query optimization example

- Process in order of increasing freq:
  - *start with smallest set, then keep cutting further.*



This is why we keep freq in dictionary

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***Brutus*** | 2 | 4 | 8 | 16 | 32 | 64 | 128 | |
| ***Calpurnia*** | 1 | 2 | 3 | 5 | 8 | 16 | 21 | 34 |
| ***Caesar*** | 13 | 16 | | | | | | |

Execute the query as (***Caesar*** *AND* ***Brutus)*** *AND* ***Calpurnia***.

Courtesy of Manning and Raghavan

# More general optimization

- e.g., *(madding OR crowd) AND (ignoble OR strife)*
- Get freq’s for all terms.
- Estimate the size of each *OR* by the sum of its freq’s (conservative).
- Process in increasing order of *OR* sizes.

Courtesy of Manning and Raghavan

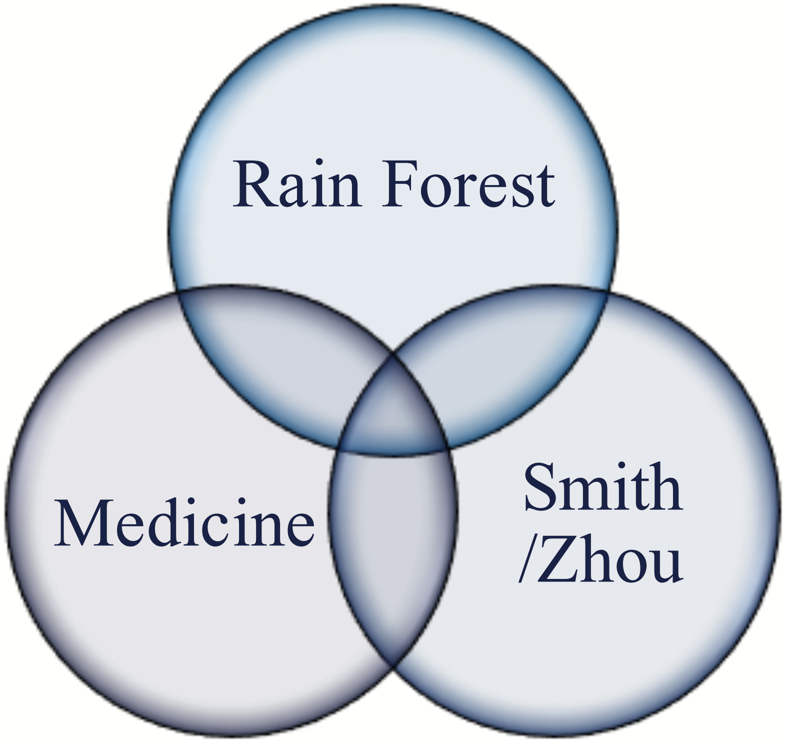
# Faceted Boolean Queries

- Strategy for forming queries: break query into facets, conjunction of disjunctions
  - each facet expresses a topic

*("rain forest" OR jungle OR amazon)* AND

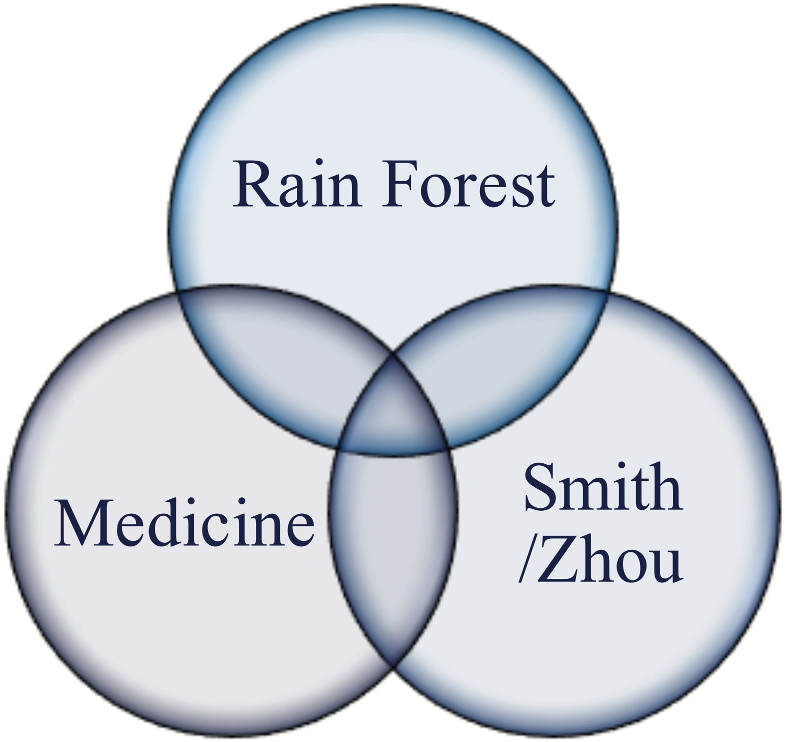
*(medicine OR remedy OR cure)* AND

*(Smith OR Zhou)*



# Faceted Boolean Query

- Query still fails if one facet missing
- Alternative:
  - Coordination level ranking
  - Order results in terms of how many facets (disjuncts) are satisfied



# Boolean Summary - Pros

- Good performance with well-constructed queries
  - ~25% more accurate on human constructed queries than an automatic non-Boolean model
- Representation is space-compact
- Bit-operations are efficient
- Results are transparent
  - Docs contain, or do not contain terms of interest
  - Semantics are well-defined

# Boolean Summary - Negatives

- If a document contains words more than once, it doesn’t matter
- If a document contains many other words besides the query terms, (is unfocused), the model ignores this
- Scores are 0/1 (specificity is low)
- Long/Complex queries are hard to construct
  - All words for concept ‘murder weapon’
  - knife or gun or hammer or sword or bow-and-arrow or rope or candlestick or poison-dart or ...

# Outline

- Course Pragmatics
  - Schedule of topics
  - Grading policy
- Overview of Text Retrieval
- Boolean Models
  - Queries
  - **Document Representations**
- Tokenization

# Amortizing Retrieval Costs

- Concerned with the *organization*, storage, and *retrieval* of textual data
- Building a document index involves an up-front cost that provides spatial and query-processing efficiencies in the large
- Libraries, the brick and mortar kind, have done this in a manual way since *forever*
  - card catalogs, classification hierarchies
- For certain individual books, manual indexes, called concordances, have been compiled.
  - Bible: Young's, Strongs, Crudens
  - Works of Shakespeare

# Term Document Matrix

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| radioactive | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| cats | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| have | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| eighteen | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| half | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| lives | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | | | | | | | | |
| … | | | | | | | | |

radioactive AND cats: D1, D7

# Term Document Matrix - Size

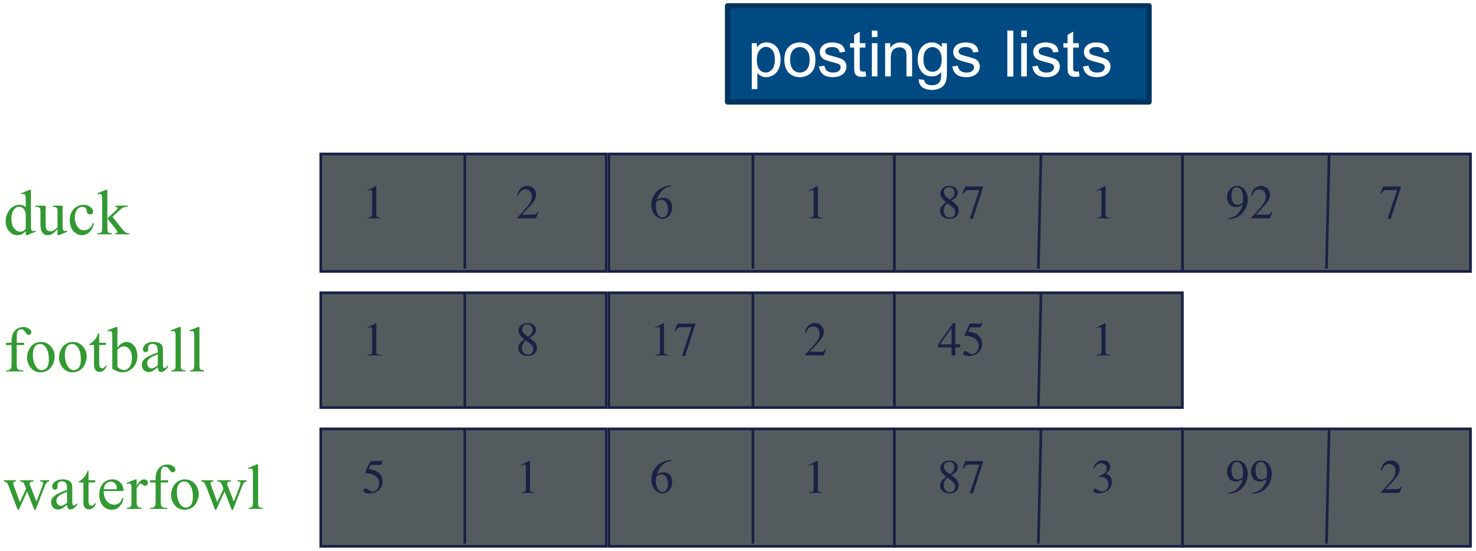
- Dense Term-Document Matrix
  - 1,000,000 terms
  - 1,000,000 documents
  - 1 bit of content (for Boolean model)

  - total: 125 GB of storage

  *(for a small collection)*

# Key Data Structure: Inverted Files

- Inverted files are a data structure that stores for each term, a list of documents containing that term
- Commonly include the number of times that term occurs; possibly even the word-order
  - Large binary files, typically 15-20% the size of the indexed text



postings lists

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| duck | 1 | 2 | 6 | 1 | 87 | 1 | 92 | 7 |
| football | 1 | 8 | 17 | 2 | 45 | 1 | | |
| waterfowl | 5 | 1 | 6 | 1 | 87 | 3 | 99 | 2 |

# Key Data Structure: Inverted Files

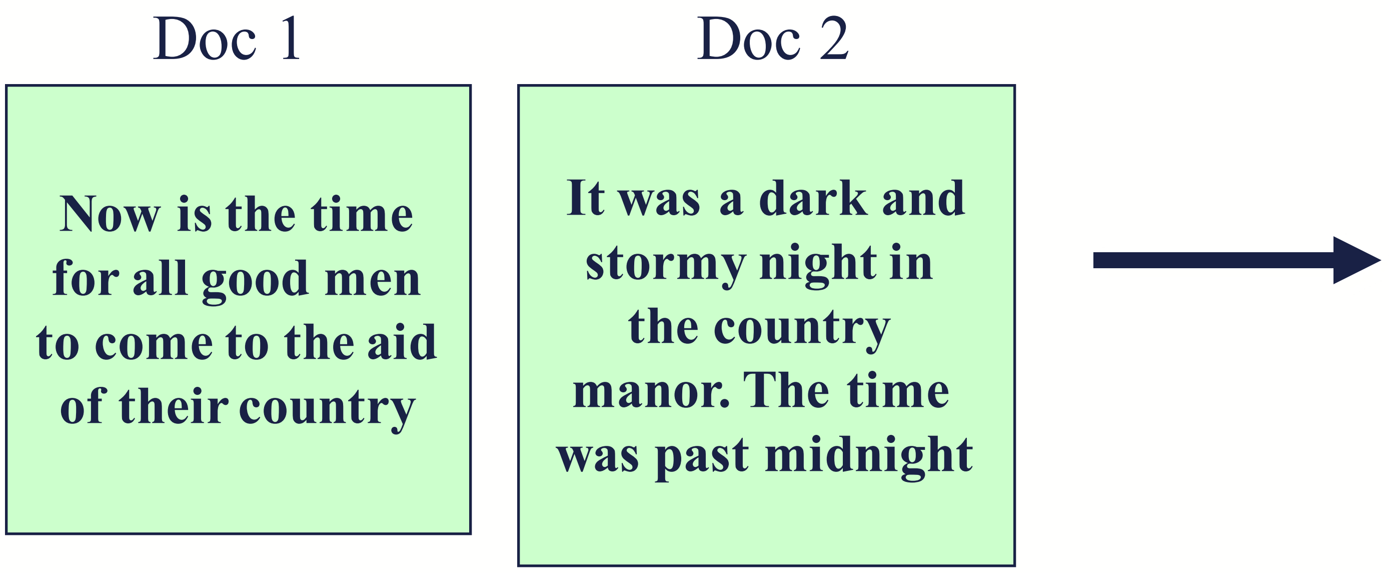
- Inverted files are a data structure that stores for each term, a list of documents containing that term
- Commonly include the number of times that term occurs; possibly even the word-order
  - Large binary files, typically 15-20% the size of the indexed text

| | doc | cnt | doc | cnt | doc | cnt | doc | cnt |
|---|---|---|---|---|---|---|---|---|
| duck | 1 | 2 | 6 | 1 | 87 | 1 | 92 | 1 |
| football | 1 | 8 | 17 | 2 | 45 | 1 | | |
| waterfowl | 5 | 1 | 6 | 1 | 87 | 3 | 101 | 3 |

Term waterfowl occurs in 4 documents. It occurs 3 times in document 87.

# Creating Inverted Files

- Documents are parsed to extract words (or stems) and these are saved with the Document ID.



Doc 1

Now is the time for all good men to come to the aid of their country

Doc 2

It was a dark and stormy night in the country manor. The time was past midnight

| Term | Doc # |
|---|---|
| now | 1 |
| is | 1 |
| the | 1 |
| time | 1 |
| for | 1 |
| all | 1 |
| good | 1 |
| men | 1 |
| to | 1 |
| come | 1 |
| to | 1 |
| the | 1 |
| aid | 1 |
| of | 1 |
| their | 1 |
| country | 1 |
| it | 2 |
| was | 2 |
| a | 2 |
| dark | 2 |
| and | 2 |
| stormy | 2 |
| night | 2 |
| in | 2 |
| the | 2 |
| country | 2 |
| manor | 2 |
| the | 2 |
| time | 2 |
| was | 2 |
| past | 2 |
| midnight | 2 |

# Creating Inverted Files

- After all document have been parsed the inverted file is sorted
- ‘Sort-based’ inversion
  - See Managing Gigabytes Section 5.2
  - IIR Chap 4

| Term | Doc # |
|---|---|
| now | 1 |
| is | 1 |
| the | 1 |
| time | 1 |
| for | 1 |
| all | 1 |
| good | 1 |
| men | 1 |
| to | 1 |
| come | 1 |
| to | 1 |
| the | 1 |
| aid | 1 |
| of | 1 |
| their | 1 |
| country | 1 |
| it | 2 |
| was | 2 |
| a | 2 |
| dark | 2 |
| and | 2 |
| stormy | 2 |
| night | 2 |
| in | 2 |
| the | 2 |
| country | 2 |
| manor | 2 |
| the | 2 |
| time | 2 |
| was | 2 |
| past | 2 |
| midnight | 2 |

→

| Term | Doc # |
|---|---|
| a | 2 |
| aid | 1 |
| all | 1 |
| and | 2 |
| come | 1 |
| country | 1 |
| country | 2 |
| dark | 2 |
| for | 1 |
| good | 1 |
| in | 2 |
| is | 1 |
| it | 2 |
| manor | 2 |
| men | 1 |
| midnight | 2 |
| night | 2 |
| now | 1 |
| of | 1 |
| past | 2 |
| stormy | 2 |
| the | 1 |
| the | 1 |
| the | 2 |
| the | 2 |
| their | 1 |
| time | 1 |
| time | 2 |
| to | 1 |
| to | 1 |
| was | 2 |
| was | 2 |

# Creating Inverted Files

- Multiple term entries for a single document are merged and frequency information added

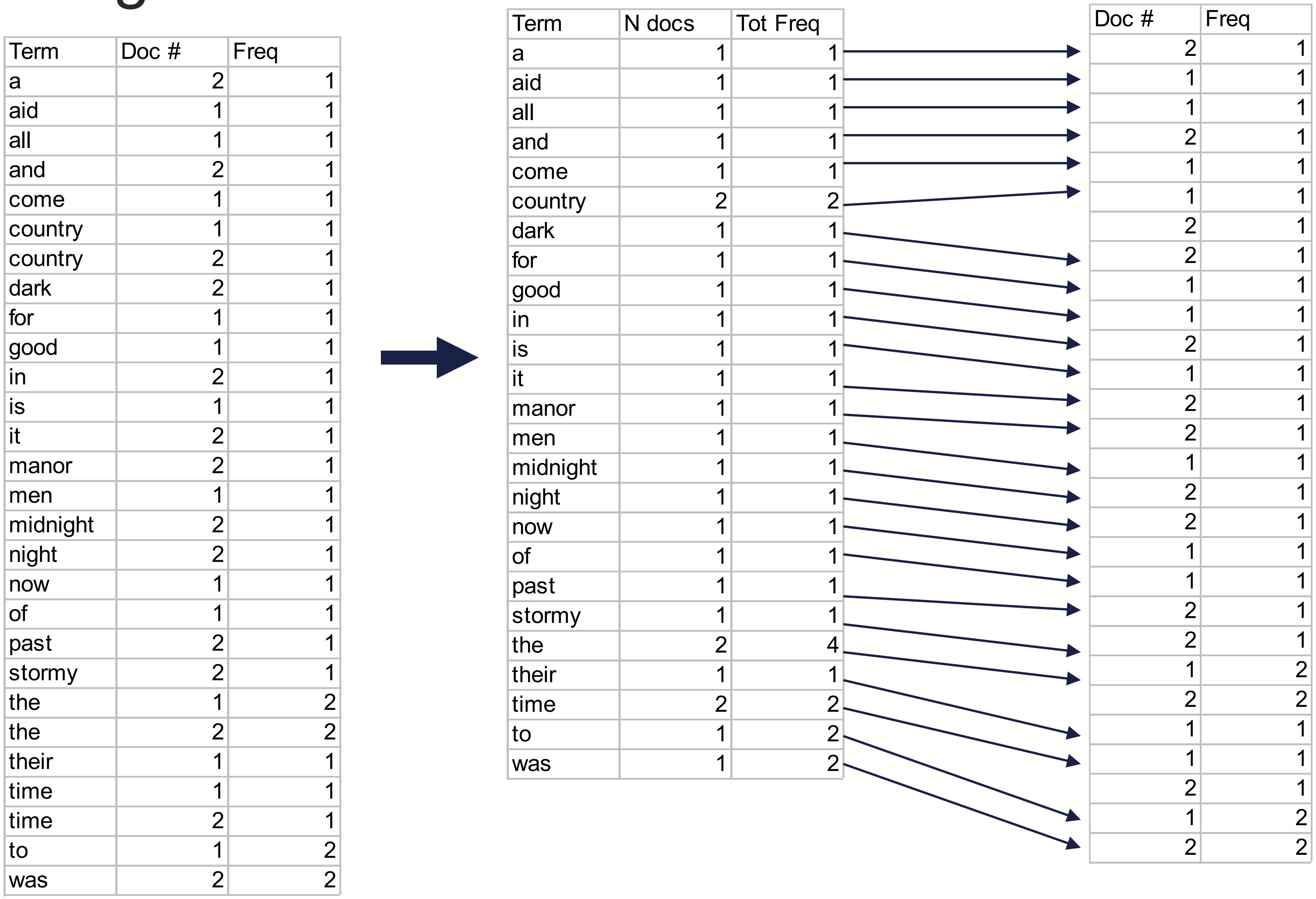
| Term | Doc # |
|---|---|
| a | 2 |
| aid | 1 |
| all | 1 |
| and | 2 |
| come | 1 |
| country | 1 |
| country | 2 |
| dark | 2 |
| for | 1 |
| good | 1 |
| in | 2 |
| is | 1 |
| it | 2 |
| manor | 2 |
| men | 1 |
| midnight | 2 |
| night | 2 |
| now | 1 |
| of | 1 |
| past | 2 |
| stormy | 2 |
| the | 1 |
| the | 1 |
| the | 2 |
| the | 2 |
| their | 1 |
| time | 1 |
| time | 2 |
| to | 1 |
| to | 1 |
| was | 2 |
| was | 2 |

→

| Term | Doc # | Freq |
|---|---|---|
| a | 2 | 1 |
| aid | 1 | 1 |
| all | 1 | 1 |
| and | 2 | 1 |
| come | 1 | 1 |
| country | 1 | 1 |
| country | 2 | 1 |
| dark | 2 | 1 |
| for | 1 | 1 |
| good | 1 | 1 |
| in | 2 | 1 |
| is | 1 | 1 |
| it | 2 | 1 |
| manor | 2 | 1 |
| men | 1 | 1 |
| midnight | 2 | 1 |
| night | 2 | 1 |
| now | 1 | 1 |
| of | 1 | 1 |
| past | 2 | 1 |
| stormy | 2 | 1 |
| the | 1 | 2 |
| the | 2 | 2 |
| their | 1 | 1 |
| time | 1 | 1 |
| time | 2 | 1 |
| to | 1 | 2 |
| was | 2 | 2 |

# Creating Inverted Files

- The file is commonly split into a Dictionary and a Postings file



| Term | Doc # | Freq |
|---|---|---|
| a | 2 | 1 |
| aid | 1 | 1 |
| all | 1 | 1 |
| and | 2 | 1 |
| come | 1 | 1 |
| country | 1 | 1 |
| country | 2 | 1 |
| dark | 2 | 1 |
| for | 1 | 1 |
| good | 1 | 1 |
| in | 2 | 1 |
| is | 1 | 1 |
| it | 2 | 1 |
| manor | 2 | 1 |
| men | 1 | 1 |
| midnight | 2 | 1 |
| night | 2 | 1 |
| now | 1 | 1 |
| of | 1 | 1 |
| past | 2 | 1 |
| stormy | 2 | 1 |
| the | 1 | 2 |
| the | 2 | 2 |
| their | 1 | 1 |
| time | 1 | 1 |
| time | 2 | 1 |
| to | 1 | 2 |
| was | 2 | 2 |

| Term | N docs | Tot Freq |
|---|---|---|
| a | 1 | 1 |
| aid | 1 | 1 |
| all | 1 | 1 |
| and | 1 | 1 |
| come | 1 | 1 |
| country | 2 | 2 |
| dark | 1 | 1 |
| for | 1 | 1 |
| good | 1 | 1 |
| in | 1 | 1 |
| is | 1 | 1 |
| it | 1 | 1 |
| manor | 1 | 1 |
| men | 1 | 1 |
| midnight | 1 | 1 |
| night | 1 | 1 |
| now | 1 | 1 |
| of | 1 | 1 |
| past | 1 | 1 |
| stormy | 1 | 1 |
| the | 2 | 4 |
| their | 1 | 1 |
| time | 2 | 2 |
| to | 1 | 2 |
| was | 1 | 2 |

| Doc # | Freq |
|---|---|
| 2 | 1 |
| 1 | 1 |
| 1 | 1 |
| 2 | 1 |
| 1 | 1 |
| 1 | 1 |
| 2 | 1 |
| 2 | 1 |
| 1 | 1 |
| 1 | 1 |
| 2 | 1 |
| 1 | 1 |
| 2 | 1 |
| 2 | 1 |
| 1 | 1 |
| 2 | 1 |
| 2 | 1 |
| 1 | 1 |
| 1 | 1 |
| 2 | 1 |
| 2 | 1 |
| 1 | 2 |
| 2 | 2 |
| 1 | 1 |
| 1 | 1 |
| 2 | 1 |
| 1 | 2 |
| 2 | 2 |

# Summary: Inverted files

- Permit fast search for individual terms

- Search results for each term is a list of document IDs (and optionally, frequency and/or positional information)

- These lists can be used to solve Boolean queries:
  - country: d1, d2
  - manor: d2
  - country and manor: d2

# Outline

- Course Pragmatics
  - Schedule of topics
  - Grading policy
- Overview of Text Retrieval
- Boolean Model
  - Queries
  - Document Representations
- **Tokenization**

# Tokenization

- Sentence Boundary Detection
- Stopwords
- Word Normalization
- Stemming
- Numbers
- Phrases

*“I Can’t Believe It’s Not Butter” is a single proper noun.*

# What is a sentence?

- Hard to find sentence boundaries
  - Structured text helps (e.g., HTML)
  - ‘.’, ‘!’, ‘?’ indicative, but ‘!’ and ‘?’ better

- 2 approaches to resolve periods
  - Knowledge based

    *Use abbreviations and knowledge of syntax*

    *Identify phone numbers, dates, email addresses*
  - Statistical classifiers built from a training corpus

- Accuracy is in the high 90s
  - 96 to 98% (see Grefenstette paper)
  - Record: 99.75% (splitta package)

# What is a word?

- Difficult to identify & normalize words
  - Steve Jobs or programming jobs
  - Baeza-Yates (surname)
  - … the ball was juggled.
  - Dr. Smith vs. Doctor Smith
  - Dr. Pepper, or, 'I Can't Believe It's Not Butter'
  - … was held at Bureau Dr. Shortly thereafter,
  - On Jan. 1, 2000, my computer still worked.
  - spoke to Jan. She said 1,200 will cost $40.

- Some words seem all but useless for retrieval
  - 'the', 'and', 'of', …

# Issues

- Punctuation
- Case
- Numbers
- Abbreviations
- Contractions
- Hyphens
- Diacritical marks

- Almost any approach has flaws

# Common Practice

- Punctuation
  - Use spaces to delimit words
  - Remove comma, colon, semi-colon, quotes, etc..
  - Perhaps note presence for further processing
  - Sometimes favor keeping interior punctuation
- Case
  - Reduce to all upper or all lower
  - Other options: preserve, preserve first character (fails on McNamee), identify acronyms
- Numbers
  - Throw away or retain some
  - Useful: Air Florida #90, 1/20/2009, Gateway 2000
  - Less useful: many consecutive digits

# Common Practice (2)

- Abbreviations
  - I.B.M. or IBM; Titles
  - Keep a list and pick canonical form

- Contractions, possessives
  - Remove suffix (don’t -> do and n’t)
  - Expand (don’t -> do not)
  - Leave interior quote marks alone

- Hyphens
  - Many uses. Use to split words?
  - What about dates: 2-19-2002
  - F-15, W-2, part-time

# Other popular steps

- Stopword removal
  - ‘stopping’
- Simple normalization of word forms
  - ‘stemming’
- Most systems do both
  - Neither is harmless
  - Both can be useful, but stemming is the more useful of the two

# Stopping

- Motivation
  - Reduce size of inverted index

    *With compression, this effect is minimal (4%)*
  - High frequency words have low discrimination power
- Lists exist (in English)

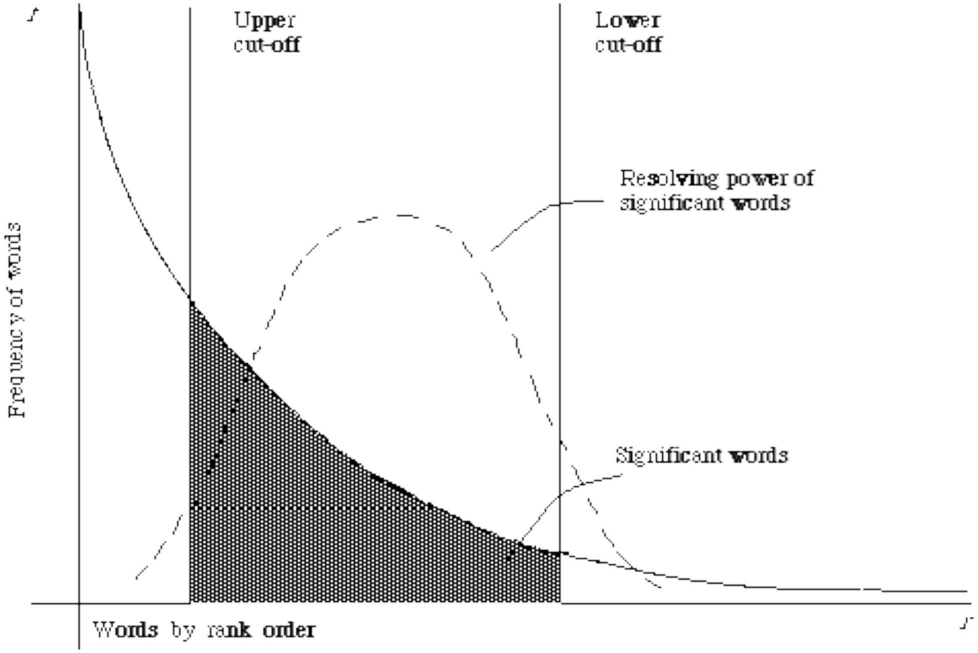


*Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)*

# Stemming

- Motivation
  - Treat word variants identically
  - Also reduce the size of the lexicon

- Example
  - remove plural forms, map cats to cat
  - juggle, juggling, juggler, juggles

    *probably shouldn't be confused with 'jug'*

    *but, suffix removal won't find jongleur*
  - physics & physician

- The technique is conflationary
  - Distinctions are lost
  - Can help and can sometimes hurt

# Morphological Analysis

- Goal: “normalize” similar words
- Morphology (“form” of words)
  - Inflectional Morphology

    *E.g,. inflect verb endings and noun number*

    *Never change grammatical class*

    - dog, dogs
    - tengo, tienes, tiene, tenemos, tienen
  - Derivational Morphology

    *Derive one word from another,*

    *Often change grammatical class*

    - build, building; health, healthy
- Problem: computationally expensive?

# Simple “S” stemming

- IF a word ends in “ies”, but not “eies” or “aies”
  - THEN “ies” → “y”
- IF a word ends in “es”, but not “aes”, “ees”, or “oes”
  - THEN “es”→ “e”
- IF a word ends in “s”, but not “us” or “ss”
  - THEN “s” → NULL

Harman, JASIS 1991

# Porter Stemmer

**Uses a list of suffixes and applies transformation rules until no further rules can be applied**

- **Multiple versions**
- **Freely available see: http://snowball.tartarus.org/**

- Too aggressive
  - organization / organ
  - policy / police
  - execute / executive
  - army / arm

- Too timid
  - european / europe
  - cylinder / cylindrical
  - create / creation
  - search / searcher

# Typical rules in Porter

- *sses* → *ss*
- *ies* → *i*
- *ational* → *ate*
- *tional* → *tion*

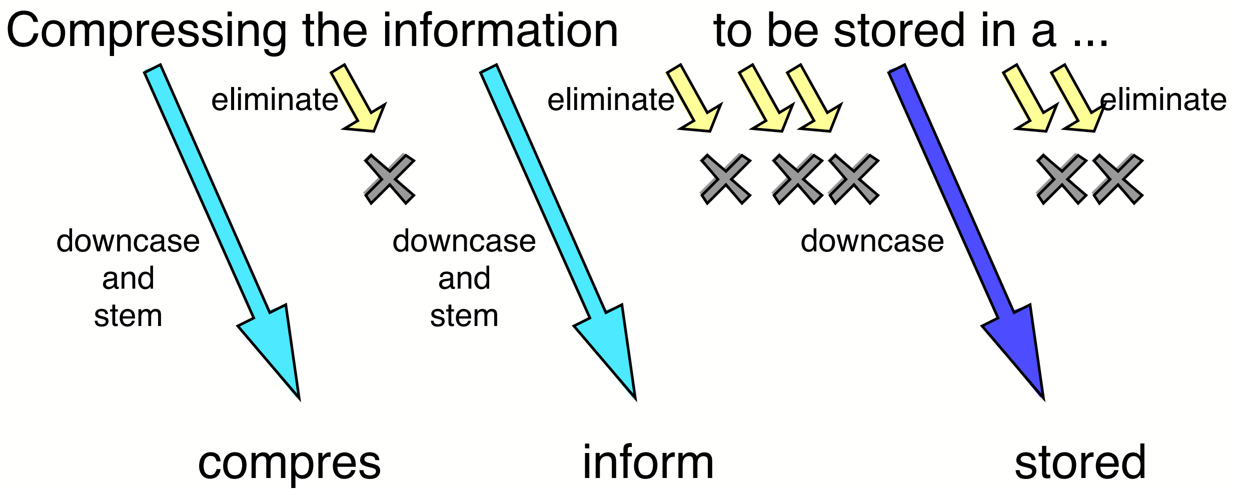
- Weight of word sensitive rules
- *(m>1) EMENT* → NULL

  *replacement* → *replac*

  *cement* → *cement*

Courtesy of Manning and Raghavan

# Representing Text



**Processing is done to both documents and queries**

# Phrases

- By far, single words are the most common unit used when representing text
- But, hard to knock intuition
  - kangaroo court
  - super bowl
  - museum of natural history
  - real estate
  - hurricane irene
- Phrase lists can be built using simple statistical methods

# Thesauri

- Some electronic thesauri exist
  - E.g., Roget’s
  - Domain specific thesauri (e.g., chemistry)

    *might map NaCl, salt, sodium chloride*
- Helpful for regional spelling differences
  - color vs. colour
- Another approach is to learn equivalences from a collection of text, statistically
- ‘Safe’ equivalences can be tricky
  - cars =?= automobiles

# Multilingual Issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - One solution is dictionary-based segmentation
- German noun compounds are not segmented
  - Lebensversicherungsgesellschaftsangestellter
  - ‘life insurance company employee’
- Schütze’s name contains an umlaut
  - Sometimes spelled Schuetze

    *common orthographic trick to avoid umlauts*
  - Sometimes the accent mark is dropped: Schutze

Derived from slides of Manning and Raghavan