Monsoon 2022

Boolean Retrieval

- Boolean Incidence matrix, Boolean queries and so on

Dr. Rajendra Prasath

Indian Institute of Information Technology Sri City, Chittoor

19th August 2022 (rajendra.2power3.com)

> Topics to be covered

- Recap:
  - IR systems
  - Classical Search Engines

- Keywords / User Information Needs
- Relevance / Irrelevance
- Personalization
- Words / Term Weighting

- Text Collection / Corpora
- Evaluation Strategy

  - More topics to come up … Stay tuned …!!

Overview

Recap: Information Retrieval

- **Information Retrieval (IR)** is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

- These days we frequently think first of web search, but there are many other cases:
  - E-mail search
  - Searching your laptop
  - Corporate knowledge bases
  - Legal information retrieval
  - and so on . . .

> Topics to be covered

- Recap:
  - Inverted Index Construction
  - Term - Document Matrix

- Boolean Operators
- Boolean Retrieval
- Boolean Queries

- Text Collection / Corpora
- Evaluation Strategy

  - More topics to come up … Stay tuned …!!

Overview

Recap: Look at 3 documents

d_1- **Darjeeling** is a city and a municipality in the Indian state of West Bengal. It is located in the Lesser Himalayas at an elevation of 6,700 feet

d_2- **Darjeeling** is noted for its tea industry, its views of Kangchenjunga, the world's third-highest mountain, and the **Darjeeling** Himalayan Railway, a UNESCO World Heritage Site

d_3- **Darjeeling** is the headquarters of the **Darjeeling** District which has a partially autonomous status within the state of West Bengal. It is also a tourist destination in India

Terms - Documents

Terms	d_1	d_2	d_3	. . .	d_n
the	2	2	3	. . .	0
a	2	1	2	. . .	1
Darjeeling	1	2	2	. . .	0
is	2	1	2	. . .	0
of	2	1	2	. . .	0
in	2	0	0	. . .	1
and	1	1	0	. . .	0
Bengal	1	0	1	. . .	0
It	1	0	1	. . .	0
Its	0	2	0	. . .	2
state	1	0	1	. . .	0
West	1	0	1	. . .	1

NOTE: "Words" and "Terms" are interchangeably used throughout the course

Boolean Incidence Matrix

Terms	d_1	d_2	d_3	. . .	d_n
the	1	1	1	. . .	0
a	1	1	1	. . .	1
Darjeeling	1	1	1	. . .	0
is	1	1	1	. . .	0
of	1	1	1	. . .	0
in	1	0	0	. . .	1
and	1	1	0	. . .	0
Bengal	1	0	1	. . .	0
It	1	0	1	. . .	0
Its	0	1	0	. . .	1
state	1	0	1	. . .	0
West	1	0	1	. . .	1

Term-document incidence matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

1 if play contains word, 0 otherwise

Brutus AND Caesar BUT NOT Calpurnia

Incidence vectors

- For each term, we have a vector consisting of 0 / 1
- To answer query: take the vectors for **Brutus**, **Caesar** and **Calpurnia** (complemented) ➔ bitwise AND
  - 110100 AND
  - 110111 AND
  - 101111 =
  - **100100**

Query: Brutus** AND **Caesar** BUT NOT **Calpurnia

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Bigger collections

- Consider *N* = 1 million documents, each with about 1000 words
- Average 6 bytes/word including spaces/punctuation

≈ 6GB of data

- Assume that there are *M* = 500K *distinct* terms among these

Can you build the matrix?

- 500K x 1M matrix has half-a-trillion 0's and 1's.
  - Why??

- But it has no more than one billion 1's.
  - matrix is extremely sparse.

- What's a better representation?
  - We only record the 1 positions.

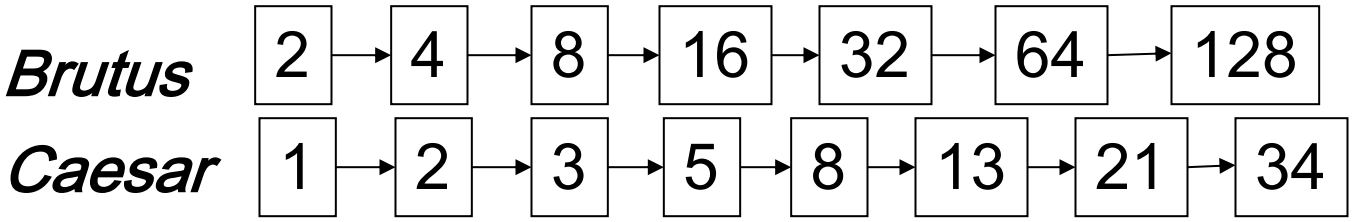
What is our focus?

- Ask for information
  - Express Information needs in terms of key words

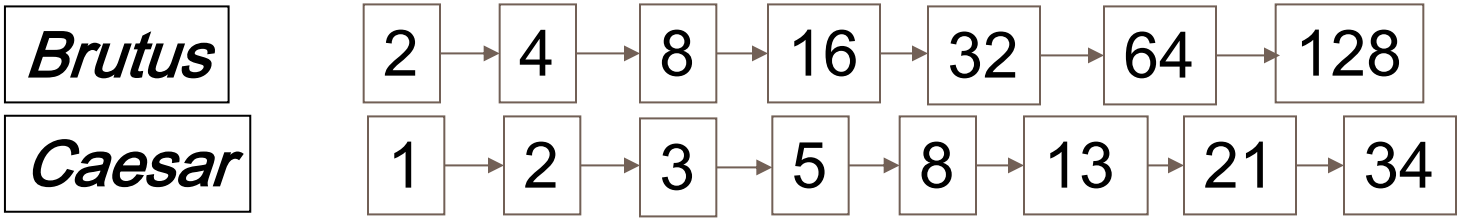
- How do we process a query?
  - Later - what kinds of queries can we process?

Query processing: AND

- Query = **Brutus** AND **Caesar**
  - Locate Brutus in the Dictionary;
    - Retrieve its postings.
  - Locate Caesar in the Dictionary;
    - Retrieve its postings.
  - "Merge" the two postings
    (intersect the document sets)



Merging of Two Postings List



- 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 Θ(x+y) operations

Crucial: postings sorted by docID.

Intersecting two postings lists (a "merge" algorithm)

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Boolean queries: Exact match

- The Boolean retrieval model is being able to ask a query that is a Boolean expression:
  - Boolean Queries are queries using *AND*, *OR* and *NOT* to join query terms
    - Views each document as a set of words
    - Is precise: document matches condition or not
  - Perhaps the simplest model to build an IR system on
- Primary commercial retrieval tool for 3 decades
- Many search systems you still use are Boolean:
  - Email, library catalog, Mac OS X Spotlight

Example: WestLaw http://www.westlaw.com/

- Largest commercial (paying subscribers) legal search service
- started in 1975; ranking added in 1992; new federated search added 2010)
- Tens of terabytes of data; ~700,000 users
- Majority of users *still* use **boolean queries**

- Example query:
  - What is the statute of limitations in cases involving the federal tort claims act?
  - LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
    - /3 = within 3 words, /S = in same sentence

Example: WestLaw http://www.westlaw.com/

- Another example query:
  - Requirements for disabled people to be able to access a workplace
- Note that SPACE is disjunction, not conjunction!
- Long, precise queries; proximity operators; incrementally developed; not like web search
- Many professional searchers still like Boolean search
  - You know exactly what you are getting

Boolean queries: More general merges

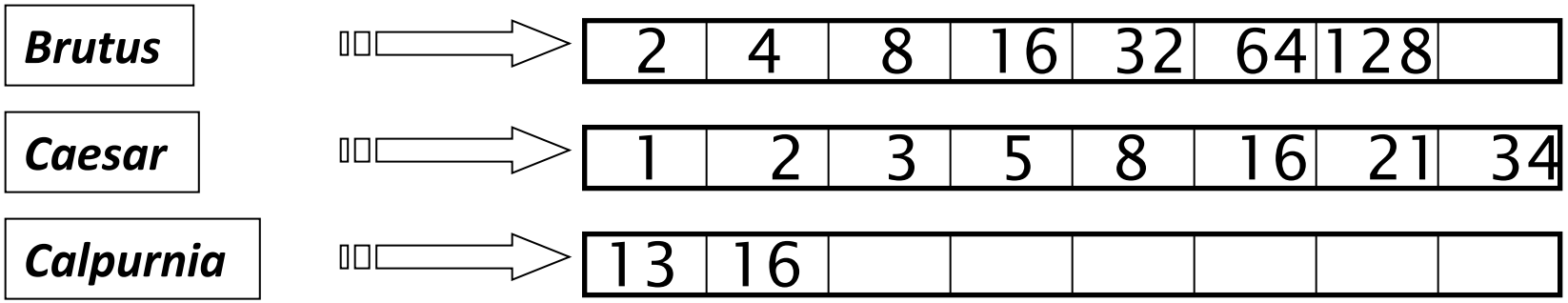
- Exercise: Adapt the merge for the queries:
  - Brutus AND NOT Caesar
  - Brutus OR NOT Caesar
- Can we still run through the merge in time $\Theta(x+y)$?
  - Linear time?
  - What can we achieve?

Merging

- What about an arbitrary Boolean formula?
- (Brutus OR Caesar) AND NOT
- (Antony OR Cleopatra)
- Can we always merge in "linear" time?
  - Linear in what?
- Can we do better?

Query optimization

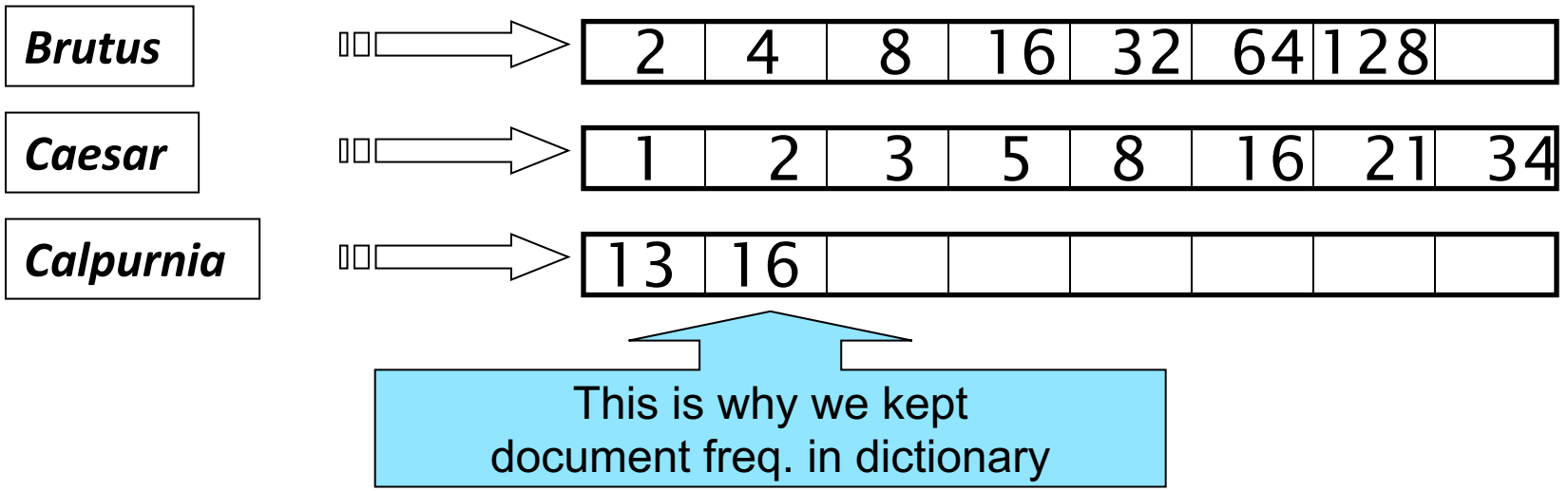
- What is the best order for query processing?
- Consider a query that is an AND of n terms.
- For each of the n terms, get its postings, then AND them together.



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

Query optimization example

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



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

More general optimization

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

Exercise

- Recommend a query processing order for

***(tangerine** OR **trees)** AND*
***(marmalade** OR **skies)** AND*
(kaleidoscope** OR **eyes)

- Which two terms should we process first?

Term	Freq
eyes	213312
kaleidosco	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

Query Processing - Exercises

- Exercise: If the query is ***friends*** *AND* ***romans*** *AND (NOT* ***countrymen****)*, how could we use the freq of ***countrymen***?
- Exercise: Extend the merge to an arbitrary Boolean query. Can we always guarantee execution in time linear in the total postings size?
- Hint: Begin with the case of a Boolean *formula* query: in this, each query term appears only once in the query

Exercise

- Try the search feature at http://www.rhymezone.com/shakespeare/
- Write down five search features you think it could do better

Summary

In this class, we focused on:

(a) Boolean Index Creation
(b) Boolean Operators
(c) Boolean Queries: AND, OR and NOT
(d) Boolean Term – Document Matrix
(e) Boolean Information Retrieval
  i. Document Retrieval
  ii. Evaluation of Boolean Retrieval
(f) Merge Algorithm
(g) Boolean Query Processing
(h) Query Optimization

Questions
It's Your Time

Contact Information

Dr. Rajendra Prasath
IIIT Sri City, Chittoor