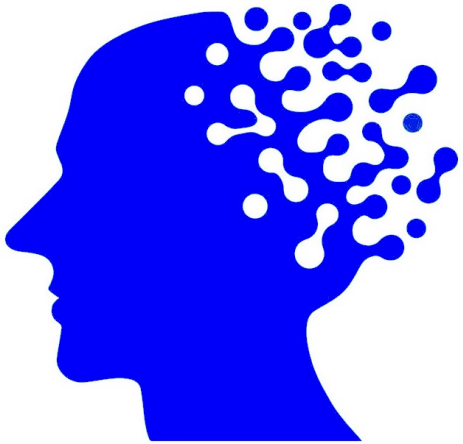


Monsoon 2021

Boolean Retrieval

- Boolean Incidence matrix, Boolean queries and so on

Dr. Rajendra Prasath

Indian Institute of Information Technology Sri City, Chittoor

24th August 2021 (rajendra.2power3.com)

> 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: 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 . . .

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

Brutus *AND* ***Caesar*** *BUT NOT* ***Calpurnia***

1 if play contains word, 0 otherwise

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.

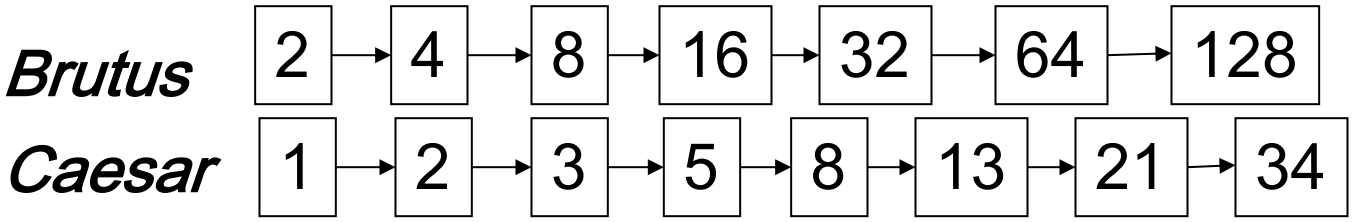
What is our focus?

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

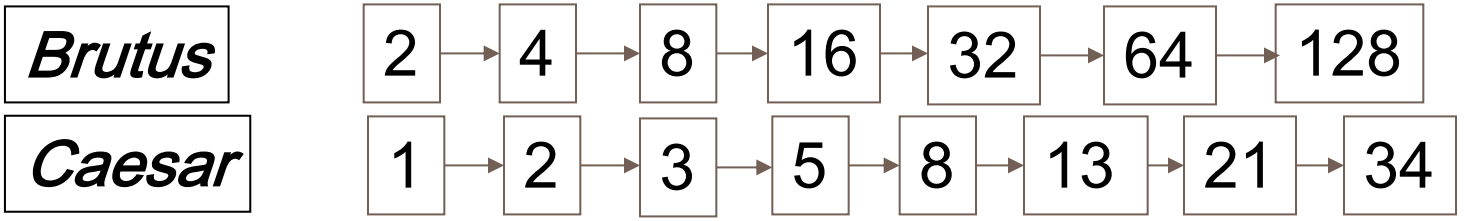
- How do we process a query?
  - Later - what kinds of queries can we process?

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
```

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 IR
    i. Document Retrieval
    ii. Evaluation of Boolean Retrieval
(f) Merge Algorithm

Questions
It's Your Time

Contact Information:

Dr. Rajendra Prasath
IIIT Sri City, Chittoor