# Basics of Information Retrieval

## Term-Document Matrix, Inverted Index and Boolean Retrieval

Debapriyo Majumdar
Indian Statistical Institute
debapriyo@isical.ac.in

# Information Retrieval

User needs some information.

An information retrieval system tries to bridge this gap.

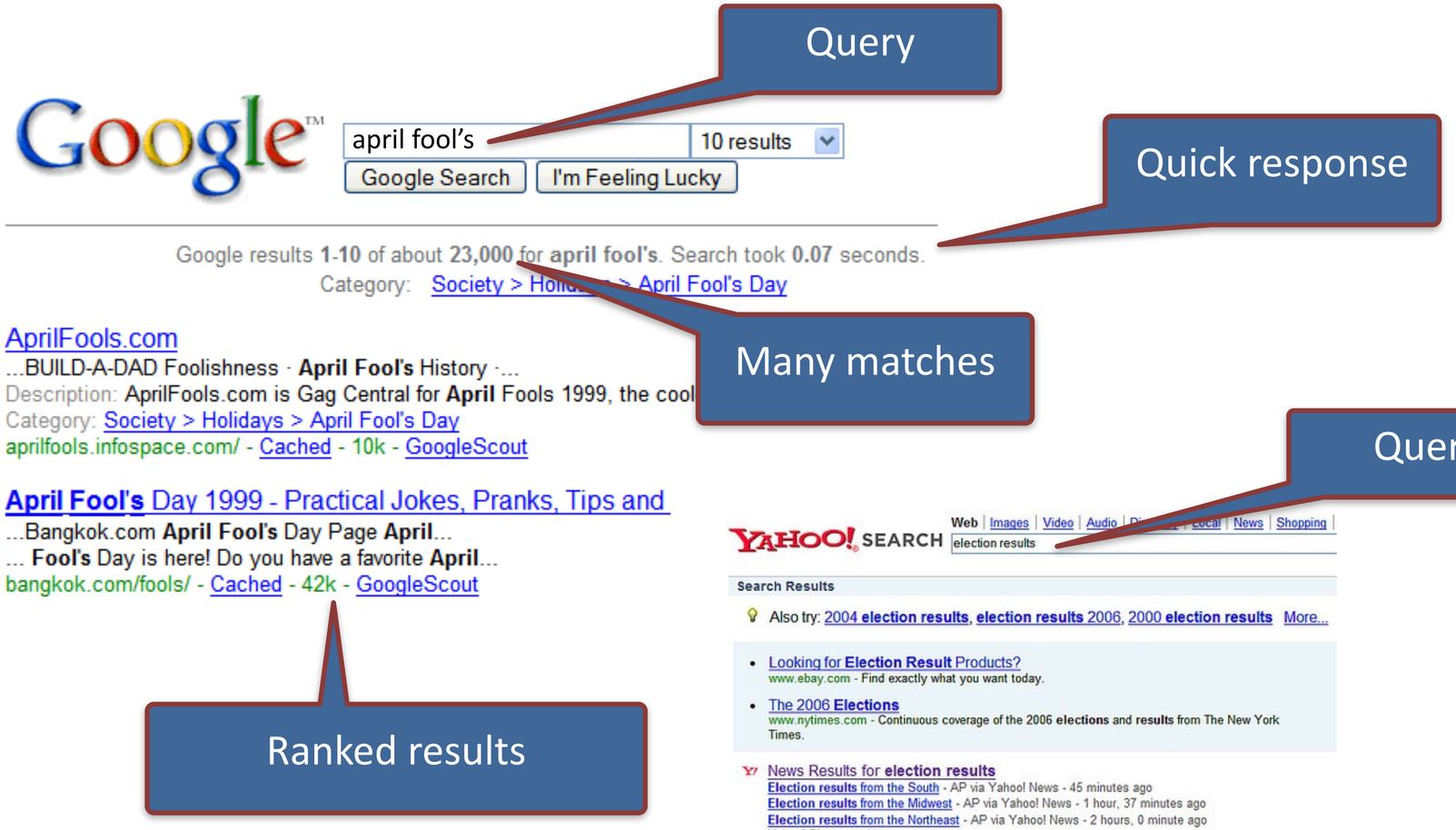
Assumption: the required information is present somewhere.

**The goal of an information retrieval system is to satisfy user's information need.**

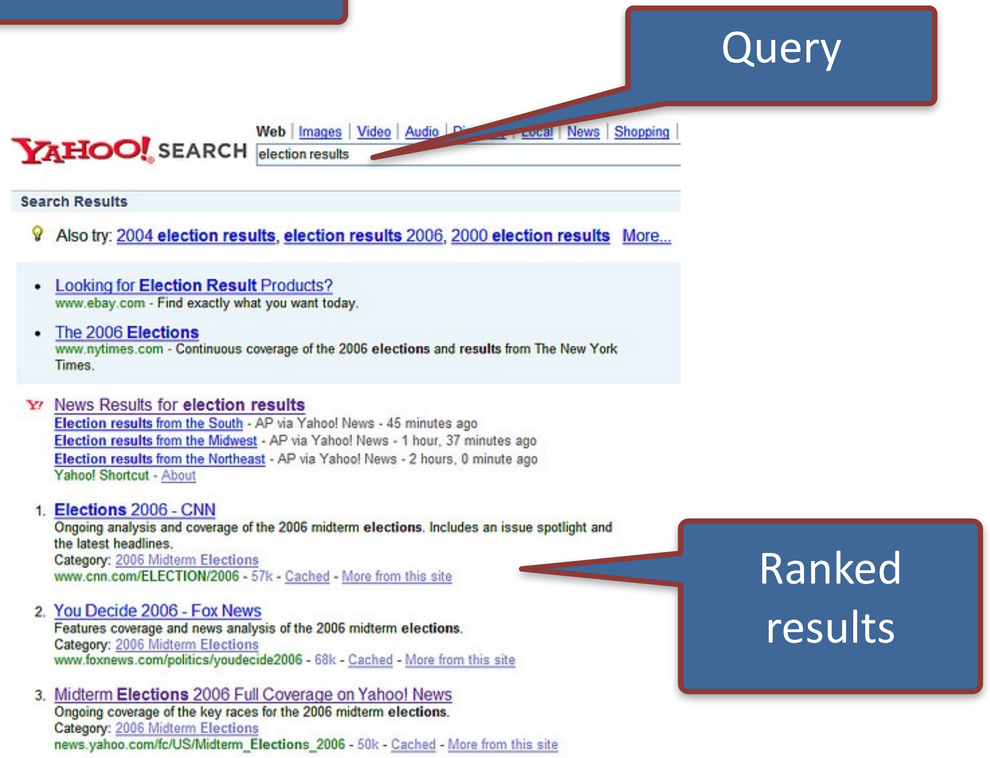
# Basic example

User expresses the information need in the form of a query.

The system returns a (ranked) list of results.




Google in 2000
Courtesy: http://googlesystem.blogspot.com/2007/12/google-in-2000.html




Yahoo search in 2006
Courtesy: https://www.searchenginewatch.com/2006/11/08/in-the-election-results-race-yahoos-the-winner/

# Collection and Documents

**The curse of the black pearl**
Ship Captain Jack Sparrow Caribbean Elizabeth Gun Fight

**Finding Nemo**
Ocean Fish Nemo Reef Animation

**Tintin**
Ocean Animation Ship Captain Haddock Tintin

**Titanic**
Ship Rose Jack Atlantic Ocean England Sink Captain

**The Dark Knight**
Bruce Wayne Batman Joker Harvey Gordon Gun Fight Crime

**Skyfall**
007 James Bond MI6 Gun Fight

**Silence of the Lambs**
Hannibal Lector FBI Crime Gun Cannibal

**The Ghost Ship**
Ship Ghost Ocean Death Horror

- Document: unit of retrieval
- Collection: the group of documents from which we retrieve
  - Also called the ***corpus*** (a body of text)

# Boolean retrieval

**The curse of the black pearl**
Ship Captain Jack Sparrow Caribbean Elizabeth Gun Fight

**Finding Nemo**
Ocean Fish Nemo Reef Animation

**Tintin**
Ocean Animation Ship Captain Haddock Tintin

**Titanic**
Ship Rose Jack Atlantic Ocean England Sink Captain

**The Dark Knight**
Bruce Wayne Batman Joker Harvey Gordon Gun Fight Crime

**Skyfall**
007 James Bond MI6 Gun Fight

**Silence of the Lambs**
Hannibal Lector FBI Crime Gun Cannibal

**The Ghost Ship**
Ship Ghost Ocean Death Horror

- Find all documents containing a word $w$
- Find all documents containing a word $w_1$ but not containing the word $w_2$
- Queries in the form of any Boolean expression
- Query: **Jack**

# Boolean retrieval

**The curse of the black pearl**
Ship Captain Jack Sparrow Caribbean Elizabeth Gun Fight

**Finding Nemo**
Ocean Fish Nemo Reef Animation

**Tintin**
Ocean Animation Ship Captain Haddock Tintin

**Titanic**
Ship Rose Jack Atlantic Ocean England Sink Captain

**The Dark Knight**
Bruce Wayne Batman Joker Harvey Gordon Gun Fight Crime

**Skyfall**
007 James Bond MI6 Gun Fight

**Silence of the Lambs**
Hannibal Lector FBI Crime Gun Cannibal

**The Ghost Ship**
Ship Ghost Ocean Death Horror

- Find all documents containing a word $w$
- Find all documents containing a word $w_1$ but not containing the word $w_2$
- Queries in the form of any Boolean expression
- Query: **Jack**

# Term – document matrix

| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Jack | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Bond | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Gun | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ocean | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Captain | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Batman | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Crime | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- The entry ($w$, $d$) = 1 if and only if the word $w$ is present in document $d$
- Terms are dimensions of this matrix (*units of index; we will discuss later*)
- Commonly called the ***term – document matrix***
- Term and word are not same, though often words are used as terms

# Boolean retrieval

| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Jack | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Bond | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Gun | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ocean | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Captain | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Batman | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Crime | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- Query: Jack
- Results: 10010000

# Boolean retrieval

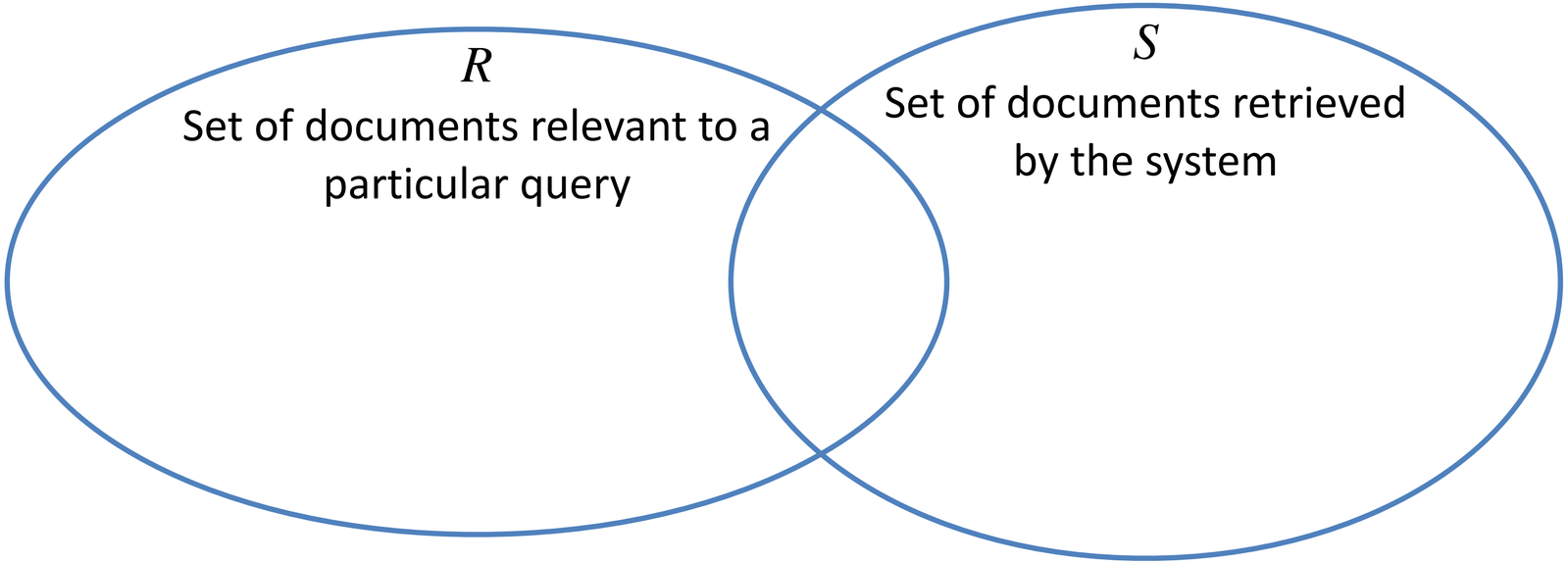
| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Jack | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Bond | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Gun | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ocean | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Captain | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Batman | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Crime | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- Query: Captain AND Gun
- Results: 10110000 && 10001110 = 10000000

# Query and relevant documents

- Query: given by user, represents the *information need*
  - Information need is the topic, conceptually what the user wants to know
  - Query is the representation of information need that the user conveys to the retrieval system
- Relevant document: a document that satisfies the information need, as perceived by the user
  - Merely matching the query terms does not mean a document is relevant
  - A relevant document must satisfy the actual information need

# Precision and recall




- What fraction of the returned results are relevant?

$$\text{Precision} = \frac{|R \cap S|}{|S|}$$

- What fraction of the relevant documents in the collection were returned by the system?

$$\text{Recall} = \frac{|R \cap S|}{|R|}$$

# What if the collection is “large”?

| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Jack | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Bond | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Gun | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ocean | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Captain | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Batman | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Crime | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

- About 1 million documents (still not so large)
- About 500,000 distinct terms
- A term – document matrix of 500,000 × 1 million Boolean entries ~ 500GB

# What if the collection is “large”?

| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Jack | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Bond | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Gun | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Ocean | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Captain | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Batman | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Crime | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

# Sparse matrix ⟶ inverted index

| | Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | | 1 | 1 | | | | 1 |
| Jack | 1 | | | 1 | | | | |
| Bond | | | | | | 1 | | |
| Gun | 1 | | | | 1 | 1 | 1 | |
| Ocean | 1 | 1 | 1 | 1 | | | | 1 |
| Captain | 1 | | 1 | 1 | | | | |
| Batman | | | | | 1 | | | |
| Crime | | | | | 1 | | 1 | |

- In reality term – document matrices are very sparse
- Most terms are NOT present in most documents
- For every term, store only the documents where the term is present

# Sparse matrix ⟶ inverted index

| | 1 Black pearl | 2 Finding Nemo | 3 Tintin | 4 Titanic | 5 Dark Knight | 6 Skyfall | 7 Silence of lambs | 8 Ghost ship |
|---|---|---|---|---|---|---|---|---|
| Ship | 1 | | 1 | 1 | | | | 1 |
| Jack | 1 | | | 1 | | | | |
| Bond | | | | | | 1 | | |
| Gun | 1 | | | | 1 | 1 | 1 | |
| Ocean | 1 | 1 | 1 | 1 | | | | 1 |
| Captain | 1 | | 1 | 1 | | | | |
| Batman | | | | | 1 | | | |
| Crime | | | | | 1 | | 1 | |

- Represent documents by document IDs

# Sparse matrix ⟶ inverted index

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |

| |
|---|
| Ship |
| Jack |
| Bond |
| Gun |
| Ocean |
| Captain |
| Batman |
| Crime |

# Sparse matrix ⟶ inverted index

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |

| | |
|---|---|
| Ship | 1, 3, 4, 8 |
| Jack | |
| Bond | |
| Gun | |
| Ocean | |
| Captain | |
| Batman | |
| Crime | |

# Sparse matrix ⟶ inverted index

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| Black pearl | Finding Nemo | Tintin | Titanic | Dark Knight | Skyfall | Silence of lambs | Ghost ship |

**Inverted Index**

| | | | | | |
|---|---|---|---|---|---|
| Ship | 1 | 3 | 4 | 8 | |
| Jack | 1 | 4 | | | |
| Bond | 6 | | | | |
| Gun | 1 | 5 | 6 | 7 | |
| Ocean | 1 | 2 | 3 | 4 | 8 |
| Captain | 1 | 3 | 4 | | |
| Batman | 5 | | | | |
| Crime | 5 | 7 | | | |

Posting lists

Vocabulary / Dictionary

# Creating an inverted index: basic idea

**1** **curse of the black pearl**
Ship Captain Jack Sparrow Caribbean Elizabeth Gun Fight

**2** **nding Nemo**
Ocean Fish Nemo Reef Animation

**3** **Tintin**
cean Animation Ship Captain Haddock Tintin

**4** **Titanic**
Ship Rose Jack Atlantic Ocean England Sink Captain

**5** **The Dark Knight**
Bruce Wayne Batman Joker Harvey Gordon Gun Fight Crime

**6** **Skyfall**
007 James Bond MI6 Gun Fight

**7** **nce of the Lambs**
Hannibal Lector FBI Crime Gun Cannibal

**8** **The Ghost Ship**
Ship Ghost Ocean Death Horror

- For each document, write out pairs (term, docid)
- Sort by term, then group

| Term | docId |
|---|---|
| Ship | 1 |
| Captain | 1 |
| Jack | 1 |
| … | … |
| Ship | 3 |
| Tintin | 3 |
| … | … |
| Jack | 4 |
| … | … |

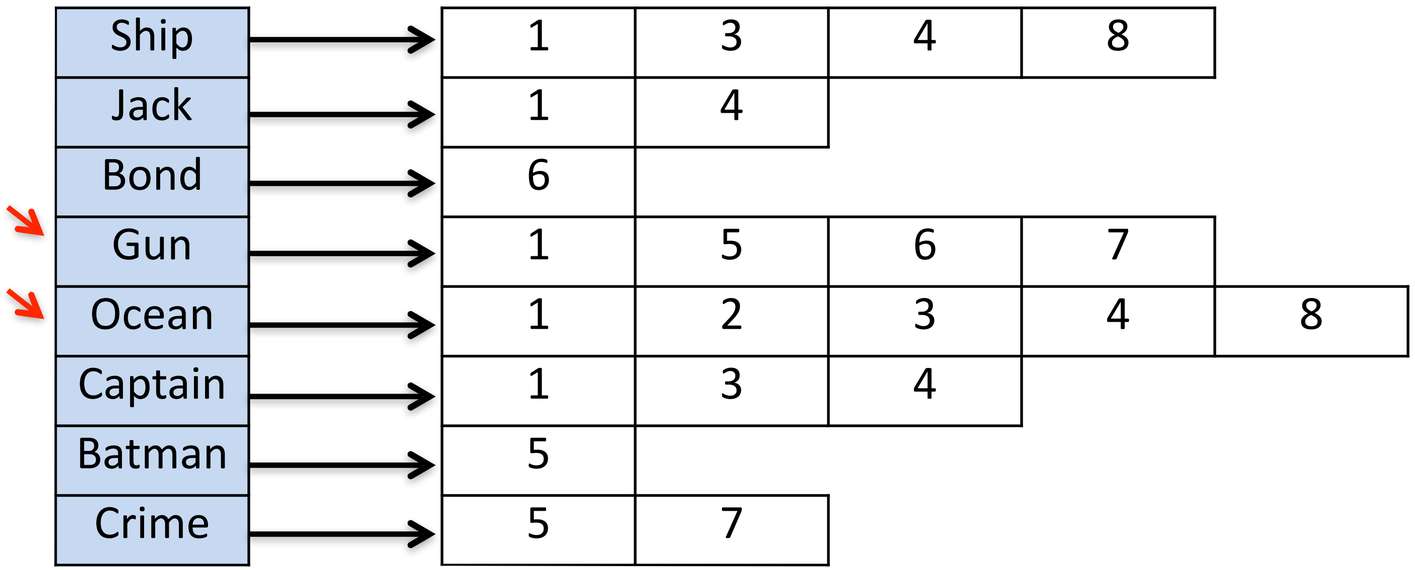
Group by term

| Term | docId |
|---|---|
| … | … |
| Captain | 1 |
| … | … |
| Jack | 1 |
| Jack | 4 |
| … | … |
| Ship | 1 |
| Ship | 3 |
| … | … |

| Term | docId | docId | docId |
|---|---|---|---|
| Captain | 1 | … | |
| Jack | 1 | 4 | … |
| Ship | 1 | 3 | … |
| … | … | | |

# Boolean query processing

**Query: Gun OR Ocean**




Need to perform **merge union** of the two lists sorted by document ID

Start with a pointer at the beginning of each list

# Boolean query processing

**Query: Gun OR Ocean**

| | | | | | |
|---|---|---|---|---|---|
| Ship | 1 | 3 | 4 | 8 | |
| Jack | 1 | 4 | | | |
| Bond | 6 | | | | |
| Gun | 1 | 5 | 6 | 7 | |
| Ocean | 1 | 2 | 3 | 4 | 8 |
| Captain | 1 | 3 | 4 | | |
| Batman | 5 | | | | |
| Crime | 5 | 7 | | | |

Both doc IDs are same $\Longrightarrow$ add to result list and advance pointers in both lists

Results: $\boxed{1}$

# Boolean query processing

**Query: Gun OR Ocean**

| | | | | | |
|---|---|---|---|---|---|
| Ship | 1 | 3 | 4 | 8 | |
| Jack | 1 | 4 | | | |
| Bond | 6 | | | | |
| Gun | 1 | 5 | 6 | 7 | |
| Ocean | 1 | 2 | 3 | 4 | 8 |
| Captain | 1 | 3 | 4 | | |
| Batman | 5 | | | | |
| Crime | 5 | 7 | | | |

Doc IDs are not same
$\Longrightarrow$ add the smaller ID
to result list and
advance only in that list

Results: 1 2

# Boolean query processing

**Query: Gun OR Ocean**

| | | | | | |
|---|---|---|---|---|---|
| Ship | 1 | 3 | 4 | 8 | |
| Jack | 1 | 4 | | | |
| Bond | 6 | | | | |
| Gun | 1 | 5 | 6 | 7 | |
| Ocean | 1 | 2 | 3 | 4 | 8 |
| Captain | 1 | 3 | 4 | | |
| Batman | 5 | | | | |
| Crime | 5 | 7 | | | |

Doc IDs are not same
$\Longrightarrow$ add the smaller ID
to result list and
advance only in that list

Results: 1 2 3

# Boolean query processing

**Query: Gun OR Ocean**

| | | | | | |
|---|---|---|---|---|---|
| Ship | 1 | 3 | 4 | 8 | |
| Jack | 1 | 4 | | | |
| Bond | 6 | | | | |
| Gun | 1 | 5 | 6 | 7 | |
| Ocean | 1 | 2 | 3 | 4 | 8 |
| Captain | 1 | 3 | 4 | | |
| Batman | 5 | | | | |
| Crime | 5 | 7 | | | |

Results: 1 2 3 4 5 6 7 8

Final result

$O(n)$ algorithm if lists are of length $O(n)$

Merge intersection works in a similar way

# Boolean retrieval use cases

Westlaw (www.westlaw.com)

- Largest commercial legal document search
- Tens of TB of text data
- Half a million users, million queries a day

Examples of queries:

- Information need: cases about a host's responsibility for drunk guests
- Example query: host! /p (responsib! liab!) /p (intoxicat! drunk!) /p guest

# References

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. "Introduction to Information Retrieval", Cambridge University Press. 2008.