# Einführung in die Informationswissenschaft

David Elsweiler| Jürgen Reischer

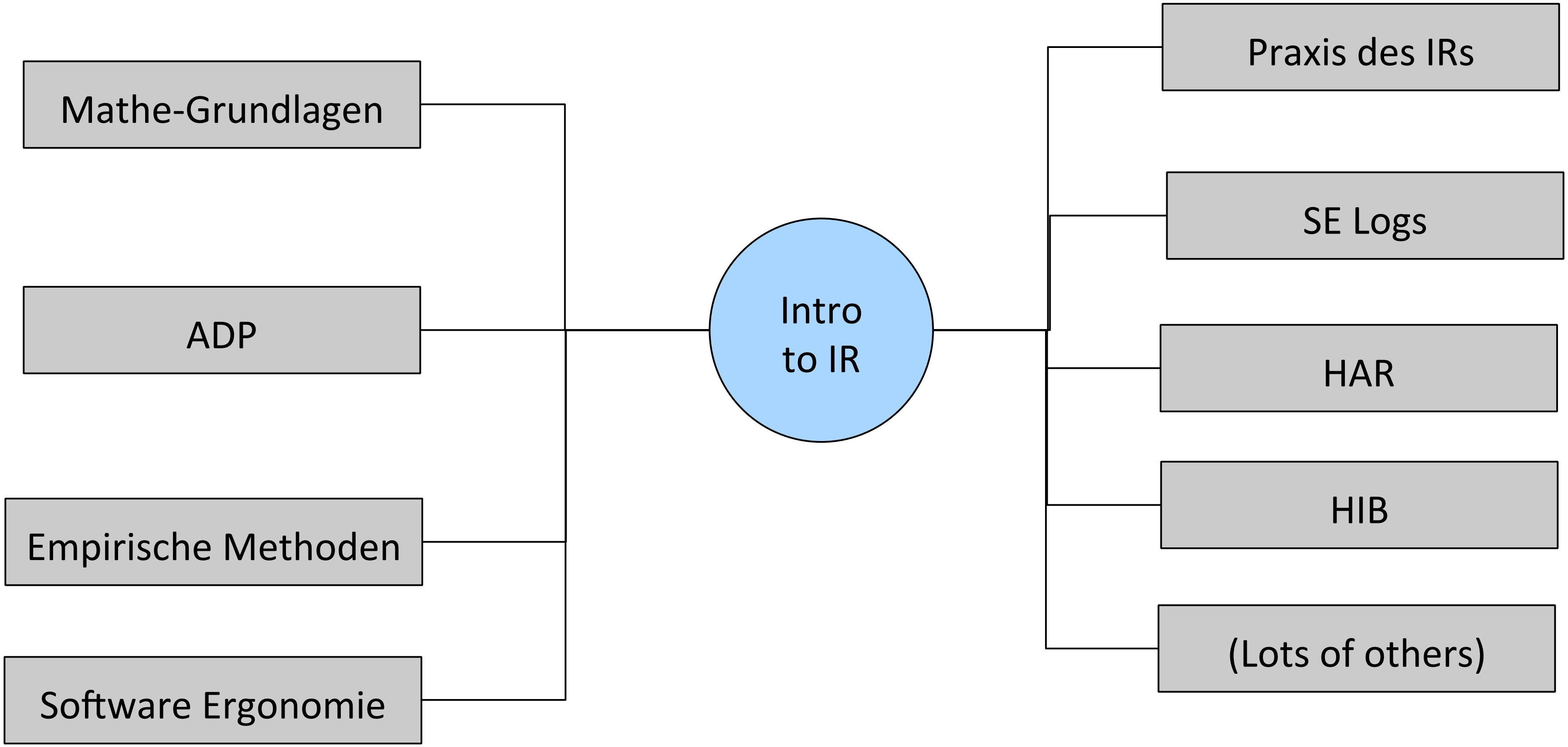
Lehrstuhl für Informationswissenschaft| www.iw.ur.de

Universität Regensburg

# At a glance

- Introduction to Information Retrieval
- Concepts involved in indexing, document pre-processing
- Boolean Retrieval

# Related Classes



Mathe-Grundlagen

ADP

Empirische Methoden

Software Ergonomie

Intro to IR

Praxis des IRs

SE Logs

HAR

HIB

(Lots of others)

# A few notes

- For those in the IR class − some of the slides might seem familiar
- For the IR class I spent a long time thinking about how to communicate the ideas (why change?)
- There is a difference:
  - Here we are just thinking about the ideas
  - Less emphasis on maths, algorithms, practicality
- The idea is to provide hooks to build on in the IR class

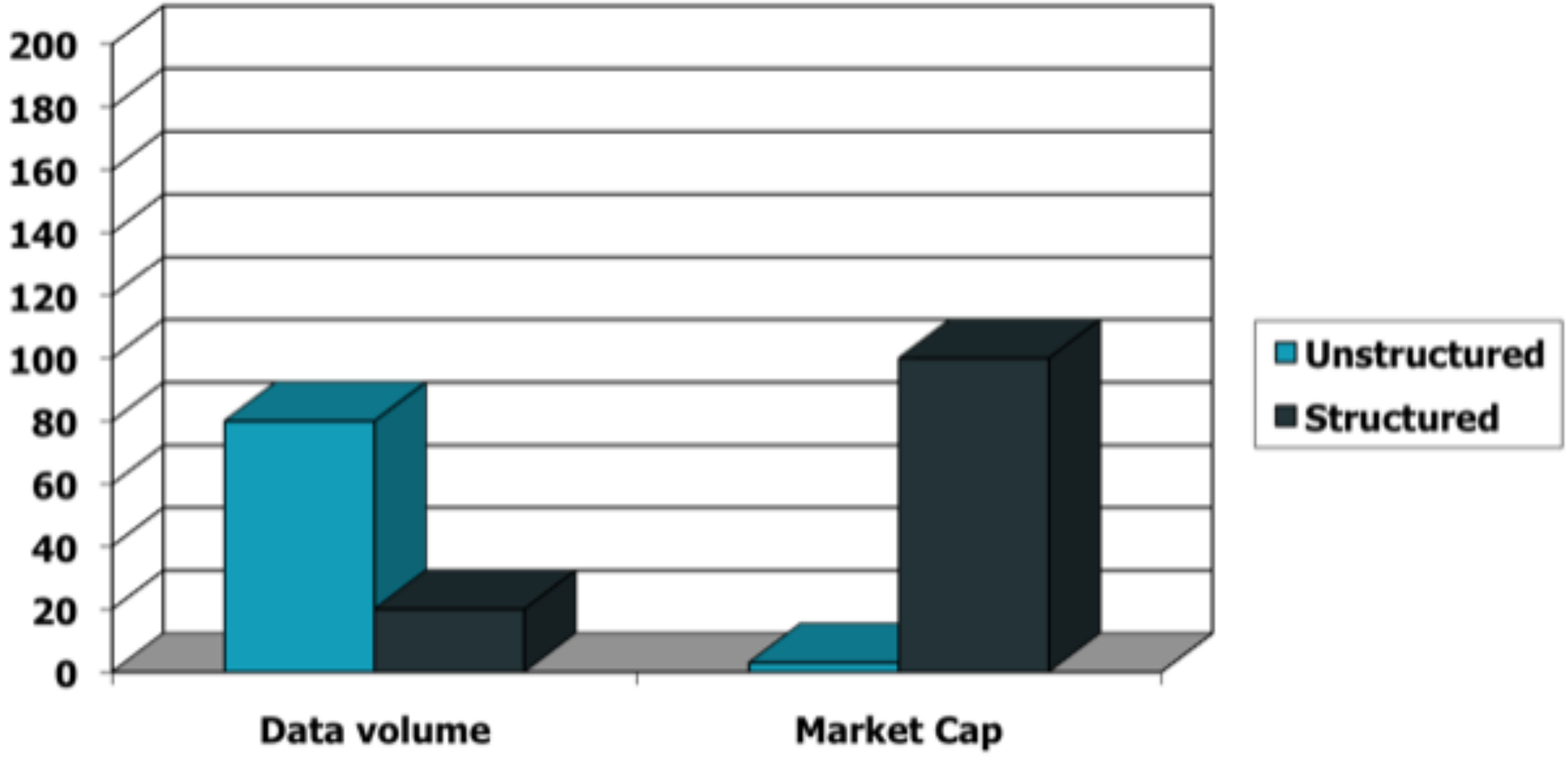
# What is information retrieval?

# #01

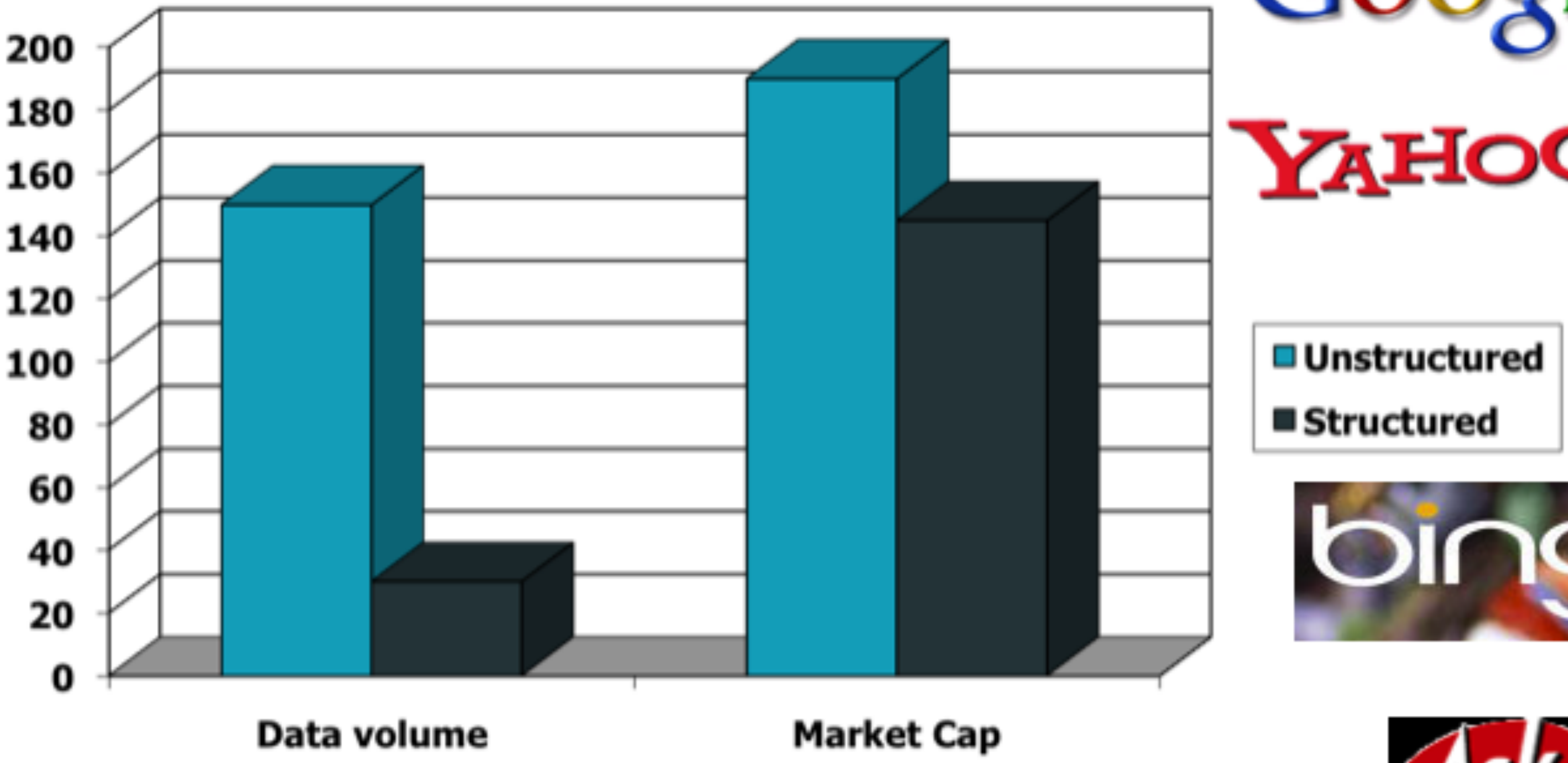
# Information Retrieval

- 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)

# Unstructured (text) vs. structured (database) data in 1996



# Unstructured (text) vs. structured (database) data in 2009



# Can you name some examples of IR Systems?

# Some Examples

- Web search
- Legal Search (Google vs Apple patent infringement)
- Desktop search
- Find the nearest pub / toilet / cinema!
- Let's discuss the differences

# Some differences

- Collection
  - Type of documents (can be mixed)
  - Size of collection (engineering differences)
  - Speed of change (in the web 100,000s new docs daily – if not more vs new pubs, new personal docs, new documents within organisations
  - Distributed vs non-distributed?

# Some differences

- Task Context
  - Work / Leisure
  - Importance (cost / implications of failure)
  - Time pressure?
  - Mobile vs non-mobile
  - How often task is performed

# Some differences

- Users
  - Experts vs novices vs mix
  - Experience level
  - Age range?
  - Care vs don't care

# Some differences

- The way queries are generated
  - Describing needs (web search)
  - From memory (desktop search)
  - Legal documents (legal search)
  - Need description + Location information (mobile search)

# Some differences

- What we might want to a system to return
  - Any relevant document (informational web search)
  - All of the relevant documents (legal)
  - *The* exact document sought-after (desktop / navigational web search)
  - Any suitable pub / all suitable pubs (mobile)

# Some differences

- Would personalisation help?
  - Web search?
  - Legal search?
  - Pub search?
  - Desktop search?

# Some differences

- How we might want to present results
    - Ordered by relevance (web)
    - Ordered by time (desktop)
    - Ordered by location (pub search)
    - What about legal search?
    - Interacting with search (facets, sorting)
    - Beyond 10 blue links ….

# Some differences

- How we might want to evaluate the system
  - Outcome (success)?
  - Speed (task completion or system response)?
  - Enjoyment?
  - Quality of things found?
  - Amount of things found?

# Evaluation

- *Precision* : Fraction of retrieved docs that are relevant to user's information need
- *Recall* : Fraction of relevant docs in collection that are retrieved
- More precise definitions and measurements to follow in later lectures

# Different Aspects

- Systems aspects
  - Technical, representation, retrieval performance, efficiency (memory and speed).
- User aspects
  - Motivation, needs, behaviour
- Interface aspects (queries, presentation, interaction)

# Implementing Information Retrieval

# #02

# We we are going to look at now

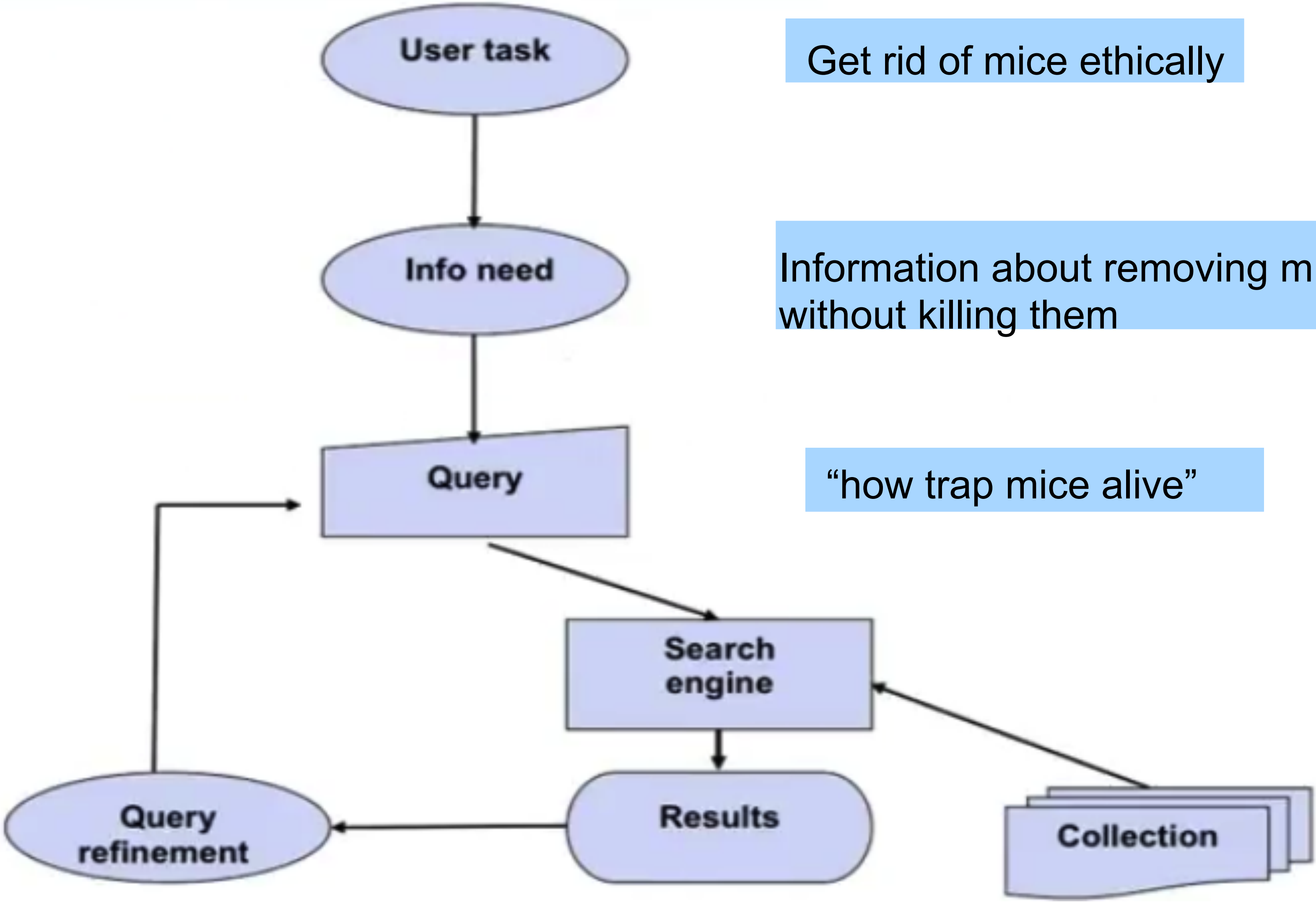
- Think about how we can represent documents in an IR system
- Think about the processing steps required
- Think about how we can use these representations to do IR

# Information Retrieval

- Collection: A set of documents
    - Assume it is static for now

- Goal: Retrieve documents with information that is relevant to the user's information need and helps the user complete a task

# The classic search model



Get rid of mice ethically

Information about removing mice without killing them

“how trap mice alive”

# Satisfying Needs

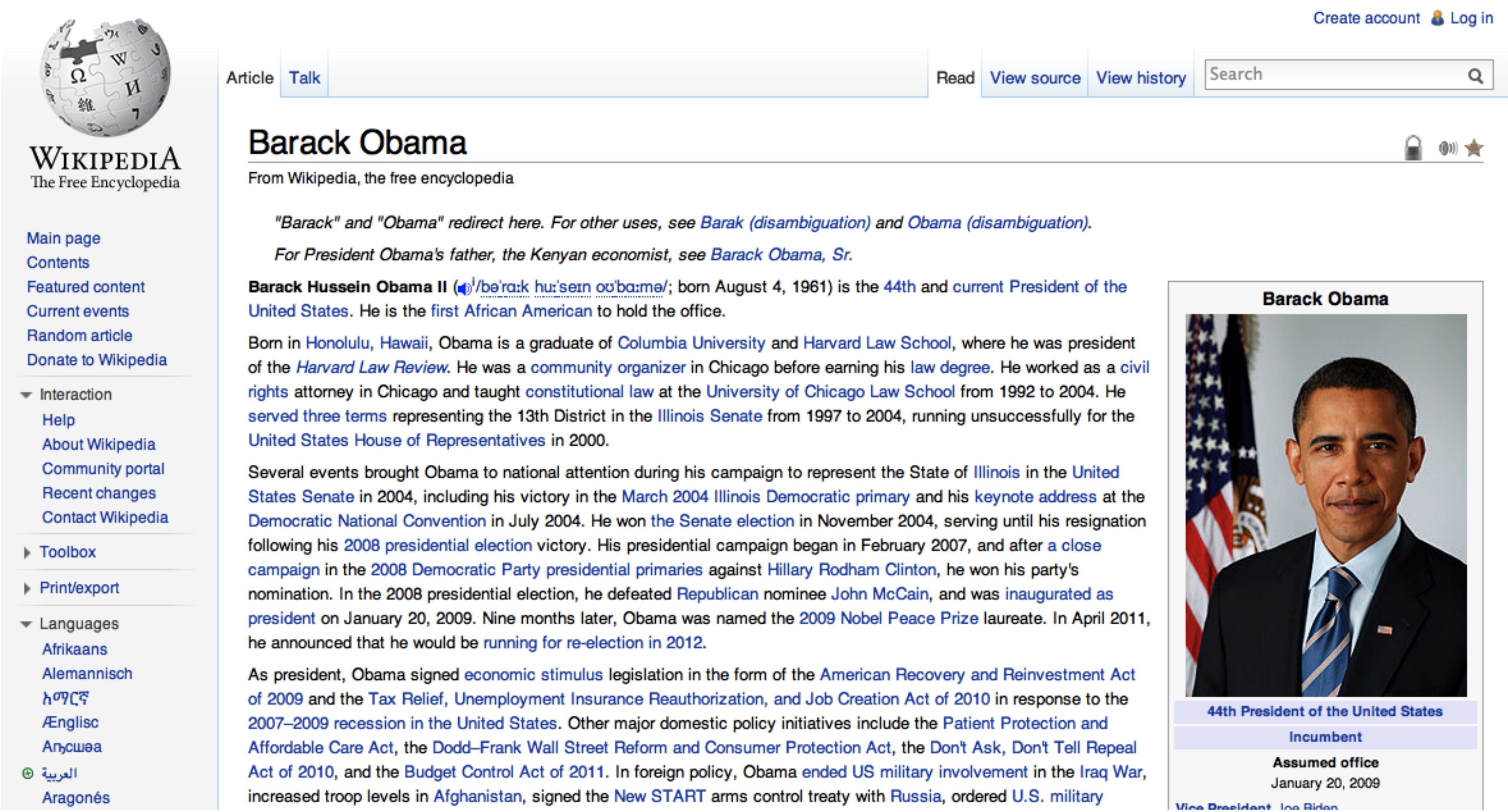
- IR Systems help people satisfy information needs
- New IR systems have lots of ways to satisfy needs (learn about these in the full IR course). Predominant form requires users to describe their needs with words

# Satisfying Needs

- Systems also represent items (docs) in collection as words (terms)
- Matching problem – we need to develop ways of calculating how similar documents and queries are
- How do we represent items, which terms do we choose?
- These questions are crucial to performance

# Representing Text

- Lots of types of documents (books, journal articles, web pages, emails, XML etc.)
- Lots of ways to represent these – we will focus on text
- Choose words that represent the content of the item  - INDEXING
- Other features interesting too (Freq. Info etc. )



# Barack Obama

From Wikipedia, the free encyclopedia

*"Barack" and "Obama" redirect here. For other uses, see Barak (disambiguation) and Obama (disambiguation).*

*For President Obama's father, the Kenyan economist, see Barack Obama, Sr.*

**Barack Hussein Obama II** (/bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is the 44th and current President of the United States. He is the first African American to hold the office.

Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was president of the *Harvard Law Review*. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney in Chicago and taught constitutional law at the University of Chicago Law School from 1992 to 2004. He served three terms representing the 13th District in the Illinois Senate from 1997 to 2004, running unsuccessfully for the United States House of Representatives in 2000.

Several events brought Obama to national attention during his campaign to represent the State of Illinois in the United States Senate in 2004, including his victory in the March 2004 Illinois Democratic primary and his keynote address at the Democratic National Convention in July 2004. He won the Senate election in November 2004, serving until his resignation following his 2008 presidential election victory. His presidential campaign began in February 2007, and after a close campaign in the 2008 Democratic Party presidential primaries against Hillary Rodham Clinton, he won his party's nomination. In the 2008 presidential election, he defeated Republican nominee John McCain, and was inaugurated as president on January 20, 2009. Nine months later, Obama was named the 2009 Nobel Peace Prize laureate. In April 2011, he announced that he would be running for re-election in 2012.

As president, Obama signed economic stimulus legislation in the form of the American Recovery and Reinvestment Act of 2009 and the Tax Relief, Unemployment Insurance Reauthorization, and Job Creation Act of 2010 in response to the 2007–2009 recession in the United States. Other major domestic policy initiatives include the Patient Protection and Affordable Care Act, the Dodd–Frank Wall Street Reform and Consumer Protection Act, the Don't Ask, Don't Tell Repeal Act of 2010, and the Budget Control Act of 2011. In foreign policy, Obama ended US military involvement in the Iraq War, increased troop levels in Afghanistan, signed the New START arms control treaty with Russia, ordered U.S. military

**Barack Obama**

**44th President of the United States**

**Incumbent**

**Assumed office**
January 20, 2009

**Vice President** Joe Biden

- Which indexing terms should we choose?

# Agreement?

- What is the agreement in class?
- Furnas et al. (1983) claim that only 10-20% of people agree on the best words to describe an item.

# Usefulness of Index Terms

- What was the first piece of legislation passed by President Obama?
- How has President Obama influenced science in the U.S.?
- What was President Obama's upbringing like?
- What is Barack Obama's middle name?

# Indexing Goals

1. Assign features that make an item easy to find given some similarity metric between doc and query
2. Assign features with enough discriminatory power that not all docs look similar to the query

# Specificity

- If we use broad (general) terms – these will apply to many documents (high recall)

- If we use narrow (precise) terms – these will only apply to few docs (high precision)

# Exhaustivity

- The more exhaustive the indexing, the more index terms per document

- How many terms do we use?
    - Also influences precision and recall
    - 1, 5, 10, …., as many terms as possible?

**Modern IR Systems use all (or most) of the terms in an item.**

**We are going to look at how!**

# Unstructured data in 1680

- Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?
- One could `grep` all of Shakespeare's plays for ***Brutus*** and ***Caesar,*** then strip out lines containing ***Calpurnia***?
- Why is that not the answer?

# Unstructured data in 1680

- Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?
- One could `grep` all of Shakespeare's plays for ***Brutus*** and ***Caesar,*** then strip out lines containing ***Calpurnia***?
- Why is that not the answer?
  - Slow (for large corpora)
  - *NOT* ***Calpurnia*** is non-trivial
  - Other operations (e.g., find the word ***Romans*** near ***countrymen***) not feasible
  - Ranked retrieval (best documents to return)
    - Later lectures

# Term / Document Matrix

# Term-document incidence

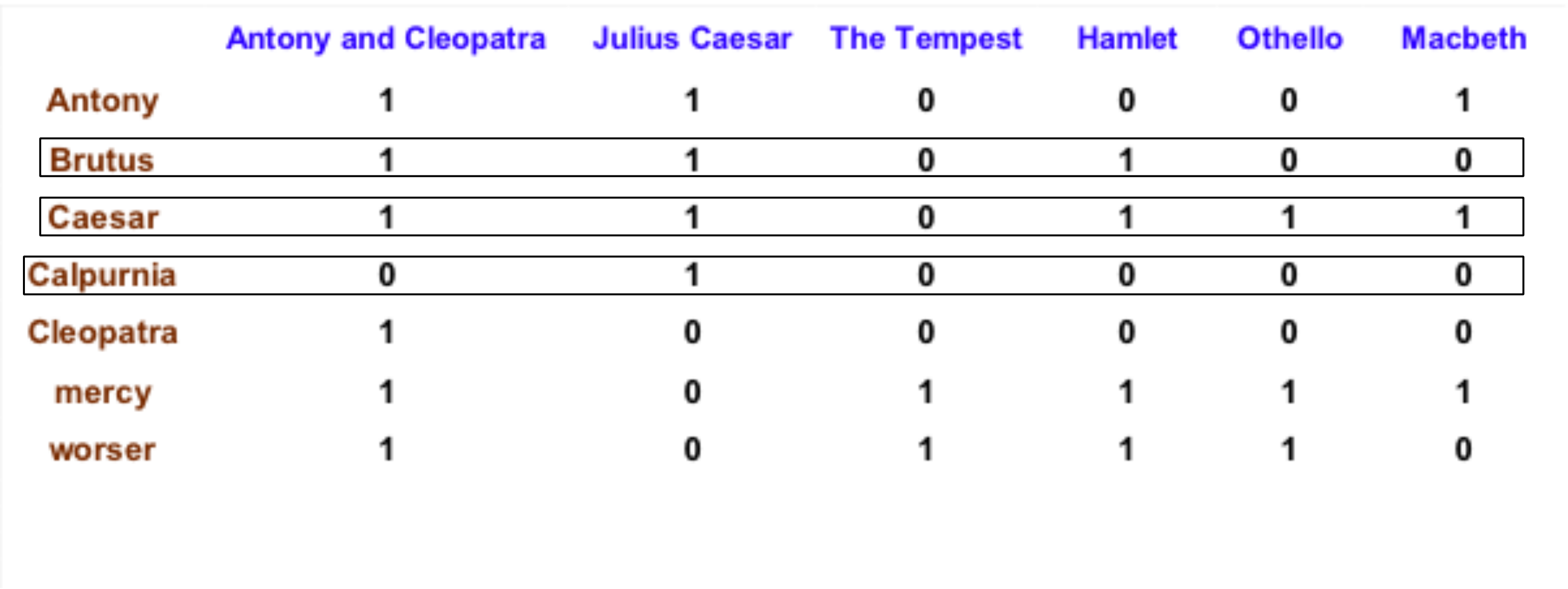
| | Antony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth |
|---|---|---|---|---|---|---|
| Antony | 1 | 1 | 0 | 0 | 0 | 1 |
| Brutus | 1 | 1 | 0 | 1 | 0 | 0 |
| Caesar | 1 | 1 | 0 | 1 | 1 | 1 |
| Calpurnia | 0 | 1 | 0 | 0 | 0 | 0 |
| Cleopatra | 1 | 0 | 0 | 0 | 0 | 0 |
| mercy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

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



| | 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 |

# Answers to query

- Antony and Cleopatra, Act III, Scene ii

*Agrippa* [Aside to DOMITIUS ENOBARBUS]: Why, Enobarbus,
When Antony found Julius ***Caesar*** dead,
He cried almost to roaring; and he wept
When at Philippi he found ***Brutus*** slain.

- Hamlet, Act III, Scene ii

*Lord Polonius:* I did enact Julius ***Caesar*** I was killed i' the
Capitol; ***Brutus*** killed me.

# A Quick Test

Given the incidence vectors for Antony, Cleopatra, and Calpurnia, i.e.
Antony: 110001
Cleopatra: 100000
Calpurnia: 010000

what is the incidence vector corresponding to the query "(Antony or Cleopatra) and not Calpurnia"?

a) 010000

b) 100001

c) 100000

d) 110001

# Bigger collections

- Consider *N* = 1 million documents, each with about 1000 words.
- Avg 6 bytes/word including spaces/punctuation
  - 6GB of data in the documents.
- Say there are *M* = 500K *distinct* terms among these.

# rows in our matrix

# We cannot build this matrix

- 500K x 1M matrix has half-a-trillion(10^12) 0's and 1's
- But it has no more than one billion 1's (10^9)
  - Matrix is extremely sparse
- What is a better representation?
  - We only record the 1 positions

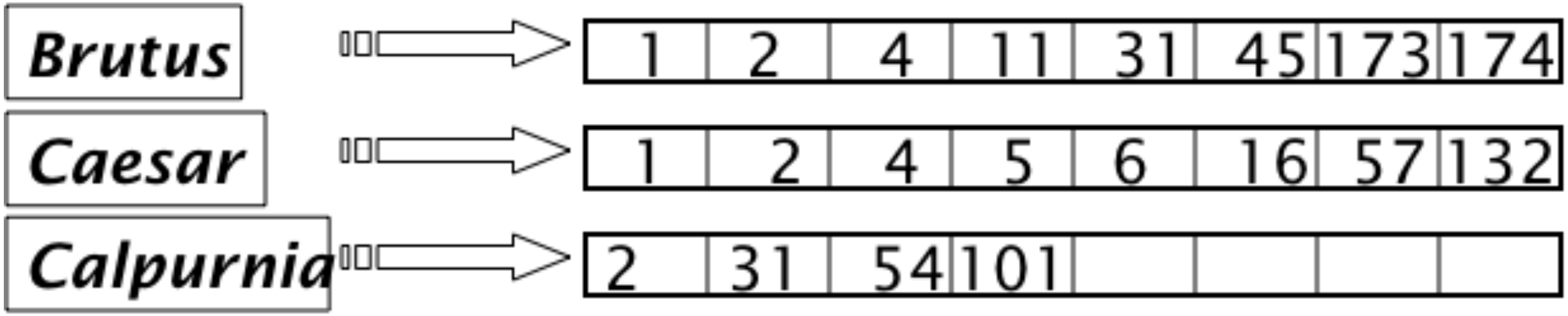
**Why?**

# Inverted Index

- Most commonly used data structure in IR
- From desktop search to huge web search engines

# Inverted index

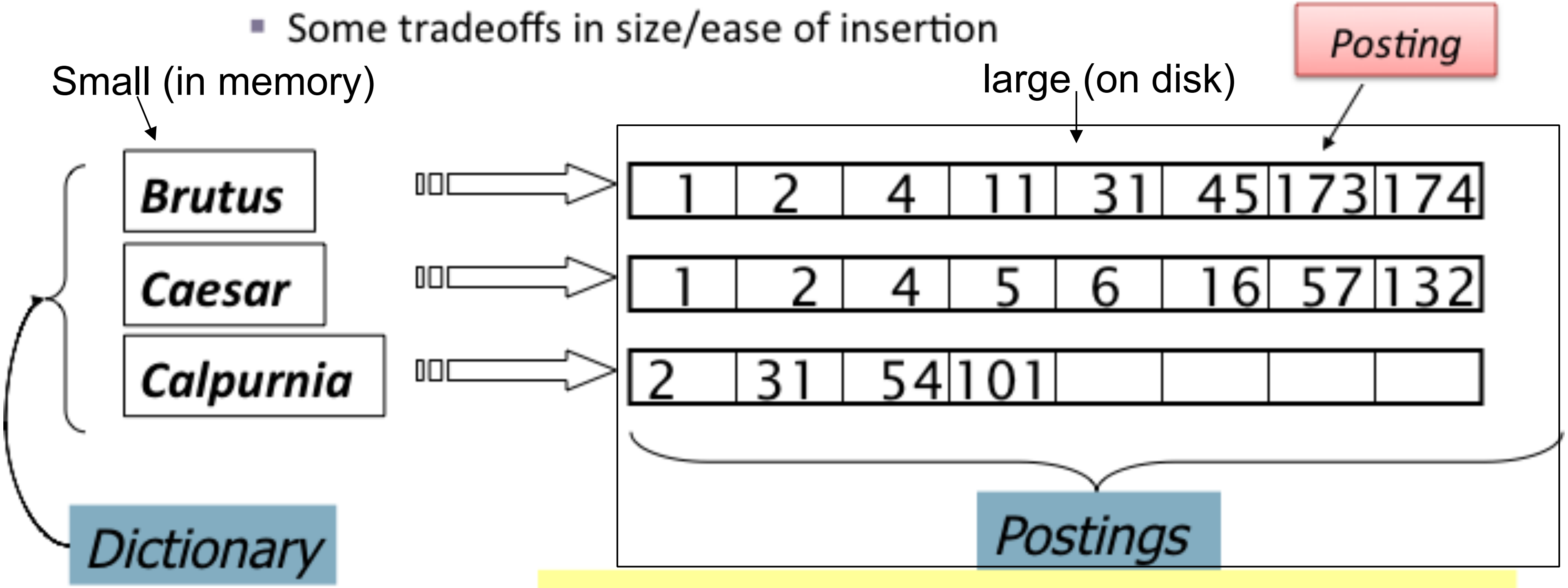
- For each term *t*, we must store a list of all documents that contain *t*.
  - Identify each by a **docID**, a document serial number
- Can we use fixed-size arrays for this?



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***Brutus*** | 1 | 2 | 4 | 11 | 31 | 45 | 173 | 174 |
| ***Caesar*** | 1 | 2 | 4 | 5 | 6 | 16 | 57 | 132 |
| ***Calpurnia*** | 2 | 31 | 54 | 101 | | | | |

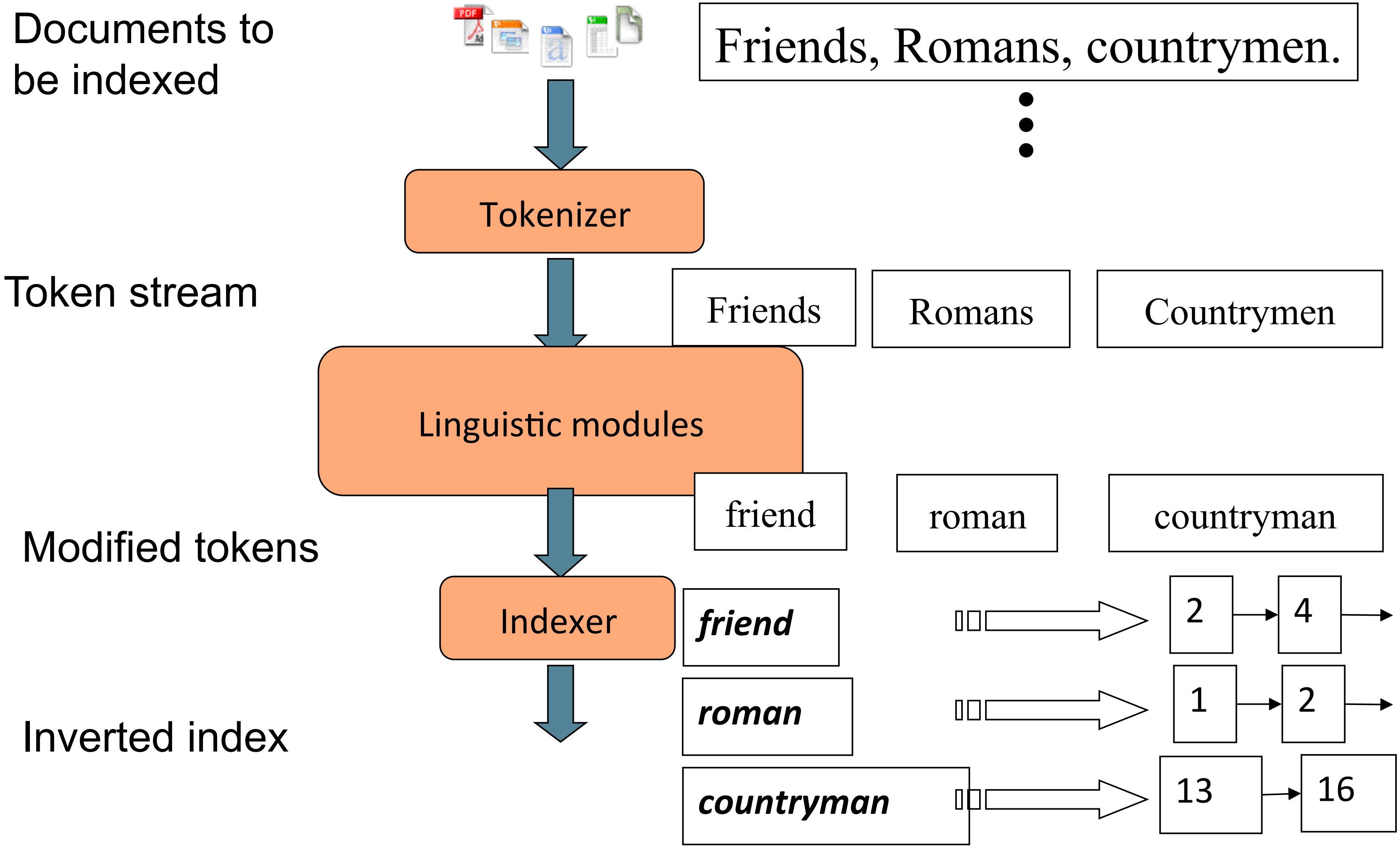
# Inverted index

- We need variable-size postings lists
  - On disk, a continuous run of postings is normal and best
  - In memory, can use linked lists or variable length arrays
    - Some tradeoffs in size/ease of insertion



Sorted by docID (more later on why).

# Inverted index construction



Documents to be indexed

Friends, Romans, countrymen.

Tokenizer

Token stream

Friends

Romans

Countrymen

Linguistic modules

Modified tokens

friend

roman

countryman

Indexer

Inverted index

*friend* → 2 → 4

*roman* → 1 → 2

*countryman* → 13 → 16

# Indexer steps: Token sequence

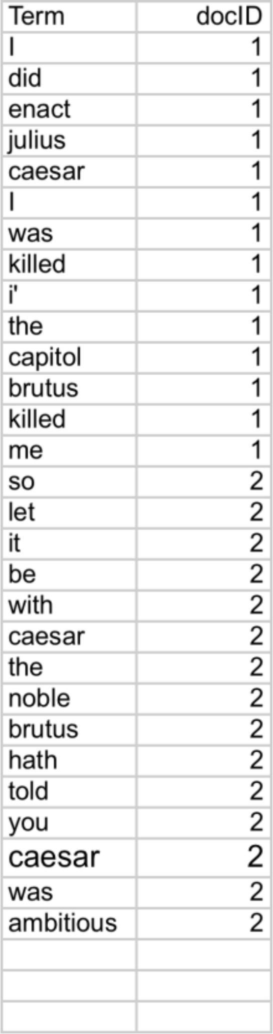
- Sequence of (Modified token, Document ID) pairs.

Doc 1

I did enact Julius
Caesar I was killed
i' the Capitol;
Brutus killed me.

Doc 2

So let it be with
Caesar. The noble
Brutus hath told you
Caesar was ambitious



| Term | docID |
|---|---|
| I | 1 |
| did | 1 |
| enact | 1 |
| julius | 1 |
| caesar | 1 |
| I | 1 |
| was | 1 |
| killed | 1 |
| i' | 1 |
| the | 1 |
| capitol | 1 |
| brutus | 1 |
| killed | 1 |
| me | 1 |
| so | 2 |
| let | 2 |
| it | 2 |
| be | 2 |
| with | 2 |
| caesar | 2 |
| the | 2 |
| noble | 2 |
| brutus | 2 |
| hath | 2 |
| told | 2 |
| you | 2 |
| caesar | 2 |
| was | 2 |
| ambitious | 2 |
| | |
| | |
| | |

# Indexer steps: Sort

- Sort by terms
  - And then docID

Core indexing step

| Term | docID |
|---|---|
| I | 1 |
| did | 1 |
| enact | 1 |
| julius | 1 |
| caesar | 1 |
| I | 1 |
| was | 1 |
| killed | 1 |
| i' | 1 |
| the | 1 |
| capitol | 1 |
| brutus | 1 |
| killed | 1 |
| me | 1 |
| so | 2 |
| let | 2 |
| it | 2 |
| be | 2 |
| with | 2 |
| caesar | 2 |
| the | 2 |
| noble | 2 |
| brutus | 2 |
| hath | 2 |
| told | 2 |
| you | 2 |
| caesar | 2 |
| was | 2 |
| ambitious | 2 |

| Term | docID |
|---|---|
| ambitious | 2 |
| be | 2 |
| brutus | 1 |
| brutus | 2 |
| capitol | 1 |
| caesar | 1 |
| caesar | 2 |
| caesar | 2 |
| did | 1 |
| enact | 1 |
| hath | 1 |
| I | 1 |
| I | 1 |
| i' | 1 |
| it | 2 |
| julius | 1 |
| killed | 1 |
| killed | 1 |
| let | 2 |
| me | 1 |
| noble | 2 |
| so | 2 |
| the | 1 |
| the | 2 |
| told | 2 |
| you | 2 |
| was | 1 |
| was | 2 |
| with | 2 |

# Indexer steps: Dictionary & Postings

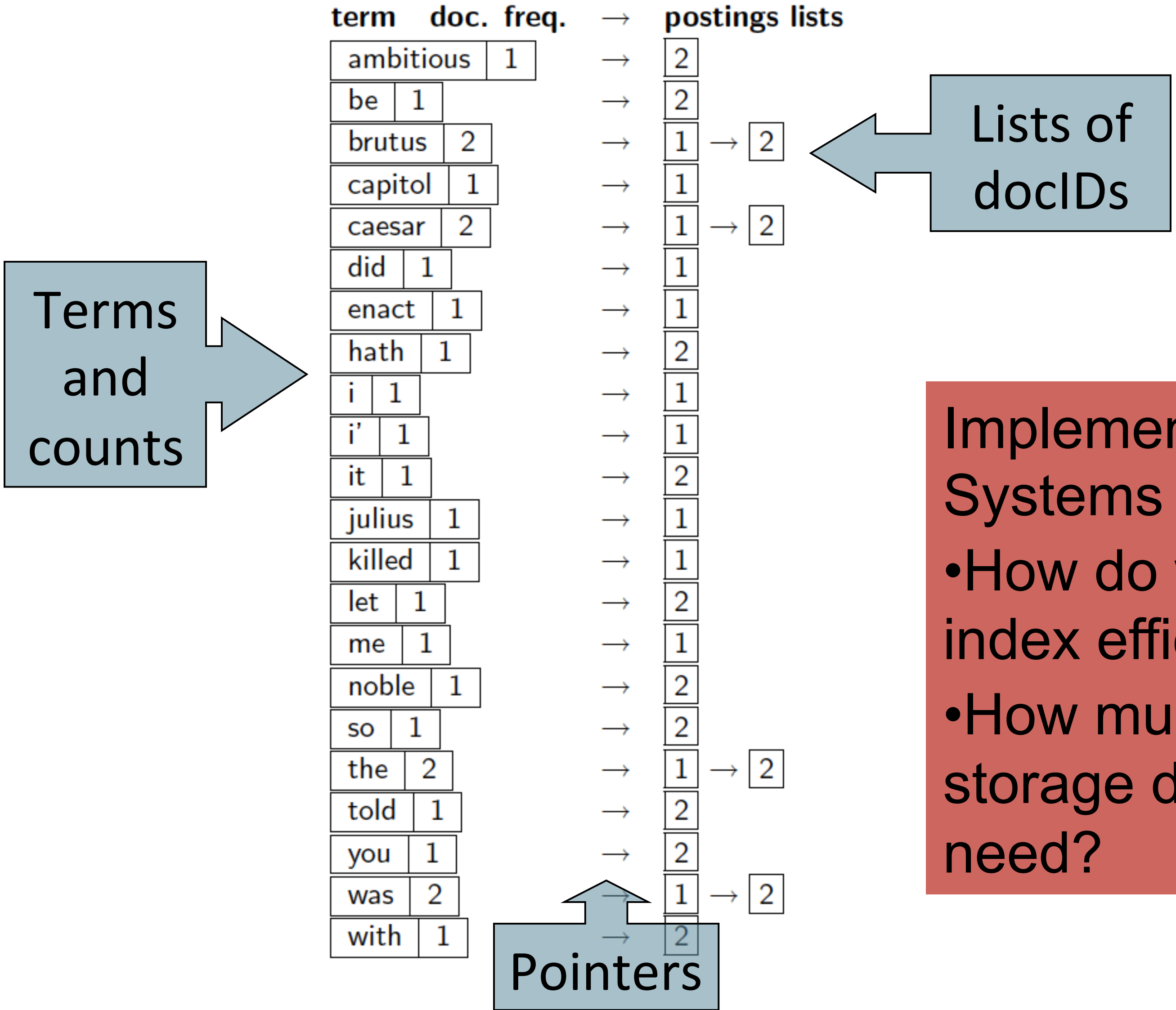
- Multiple term entries in a single document are merged.
- Split into Dictionary and Postings
- Doc. frequency information is added.

Why frequency?
Will discuss later.

| Term | docID |
|---|---|
| ambitious | 2 |
| be | 2 |
| brutus | 1 |
| brutus | 2 |
| capitol | 1 |
| caesar | 1 |
| caesar | 2 |
| caesar | 2 |
| did | 1 |
| enact | 1 |
| hath | 1 |
| I | 1 |
| I | 1 |
| i' | 1 |
| it | 2 |
| julius | 1 |
| killed | 1 |
| killed | 1 |
| let | 2 |
| me | 1 |
| noble | 2 |
| so | 2 |
| the | 1 |
| the | 2 |
| told | 2 |
| you | 2 |
| was | 1 |
| was | 2 |
| with | 2 |

| term | doc. freq. | → | postings lists |
|---|---|---|---|
| ambitious | 1 | → | 2 |
| be | 1 | → | 2 |
| brutus | 2 | → | 1 → 2 |
| capitol | 1 | → | 1 |
| caesar | 2 | → | 1 → 2 |
| did | 1 | → | 1 |
| enact | 1 | → | 1 |
| hath | 1 | → | 2 |
| i | 1 | → | 1 |
| i' | 1 | → | 1 |
| it | 1 | → | 2 |
| julius | 1 | → | 1 |
| killed | 1 | → | 1 |
| let | 1 | → | 2 |
| me | 1 | → | 1 |
| noble | 1 | → | 2 |
| so | 1 | → | 2 |
| the | 2 | → | 1 → 2 |
| told | 1 | → | 2 |
| you | 1 | → | 2 |
| was | 2 | → | 1 → 2 |
| with | 1 | → | 2 |

# Where do we pay in storage?



| term | doc. freq. | → | postings lists |
|---|---|---|---|
| ambitious | 1 | → | 2 |
| be | 1 | → | 2 |
| brutus | 2 | → | 1 → 2 |
| capitol | 1 | → | 1 |
| caesar | 2 | → | 1 → 2 |
| did | 1 | → | 1 |
| enact | 1 | → | 1 |
| hath | 1 | → | 2 |
| i | 1 | → | 1 |
| i' | 1 | → | 1 |
| it | 1 | → | 2 |
| julius | 1 | → | 1 |
| killed | 1 | → | 1 |
| let | 1 | → | 2 |
| me | 1 | → | 1 |
| noble | 1 | → | 2 |
| so | 1 | → | 2 |
| the | 2 | → | 1 → 2 |
| told | 1 | → | 2 |
| you | 1 | → | 2 |
| was | 2 | → | 1 → 2 |
| with | 1 | → | 2 |

Terms and counts

Lists of docIDs

Pointers

Implementing IR Systems

- How do we index efficiently?
- How much storage do we need?

# The index we just built

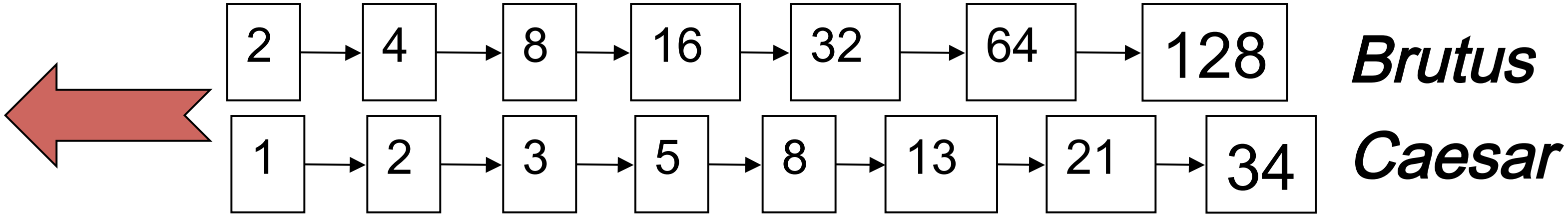
- How do we process a query?
  - Walk through the steps of processing a query using this kind of inverted index structure
  - Later – what kinds of queries can we process?

# Query processing: AND

- Consider processing the query:

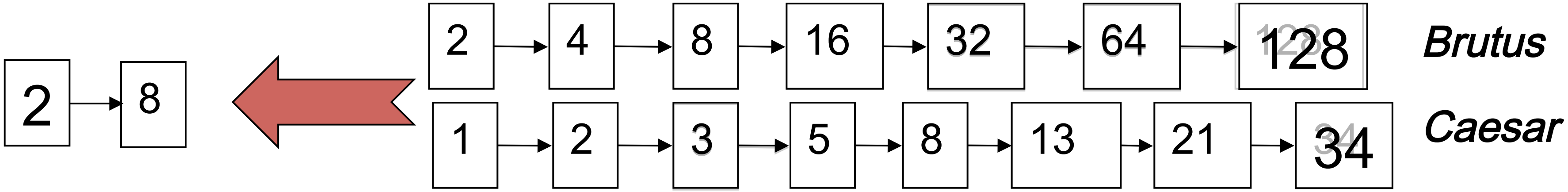
  ***Brutus*** *AND* ***Caesar***

  - Locate ***Brutus*** in the Dictionary;
    - Retrieve its postings.
  - Locate *Caesar* in the Dictionary;
    - Retrieve its postings.
  - "Merge" the two postings:



# The merge

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



If list lengths are *x* and *y*, merge takes O(*x*+*y*) operations.
Crucial: postings sorted by docID.

# Summary

# #03

# In summary

- We now know what information retrieval is and why it is important to information science
- We understand why „Grep“ is not the answer
- We introduced the concept of indexing and talked about the inverted index and processes involved in building one
- We introduced boolean retrieval. Still used today in many fields.
- Homework – think about what the limitations of boolean search might be.