COMP38120
Workshop 2: Principles of IR

John McNaught
& Sandra Sampaio

Overview

- Comparing search engines
- IR system basics
- Characterising (indexing) documents and queries
- The notion of information need
- A basic model: the Boolean model
- Basic indexing techniques
- Query processing in the Boolean model
- The extended Boolean model

Wk 2

MANCHESTER
1824
The University of Manchester

Comparing search engines (record answers for later comparison with other groups)

- In your groups
  - Determine which search engines your group has used recently
  - Determine to what extent you each found what you were looking for from these engines
    - Individually: rate each engine with 1..5 stars (5=perfect)
    - Group: agree an overall rating for engines used by >1 member

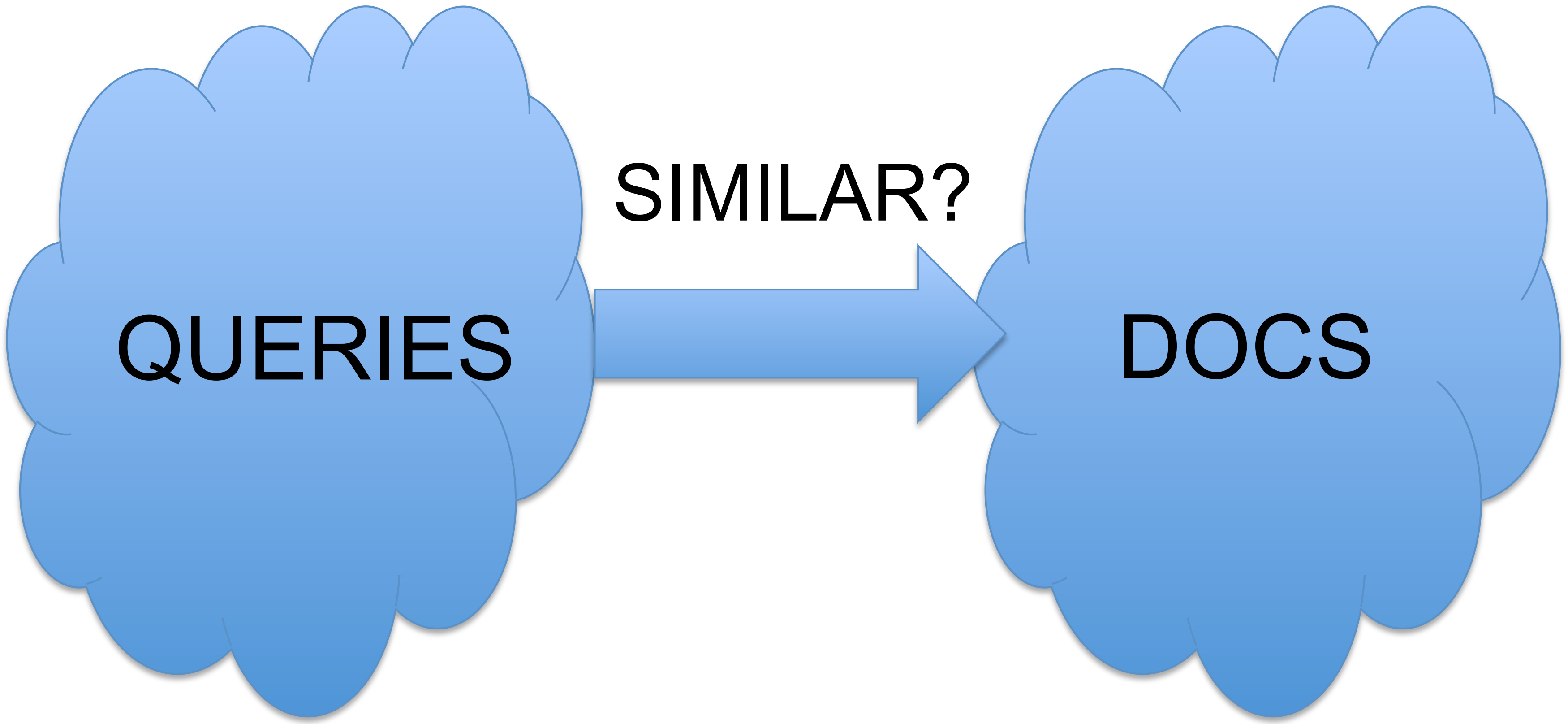
Comparing: Group questions

- What would you change to make each engine better?
- How can you be certain you found what you were looking for (if you found it)?
- How do you know you didn't miss something important?
- → report back / class discussion

The University of Manchester

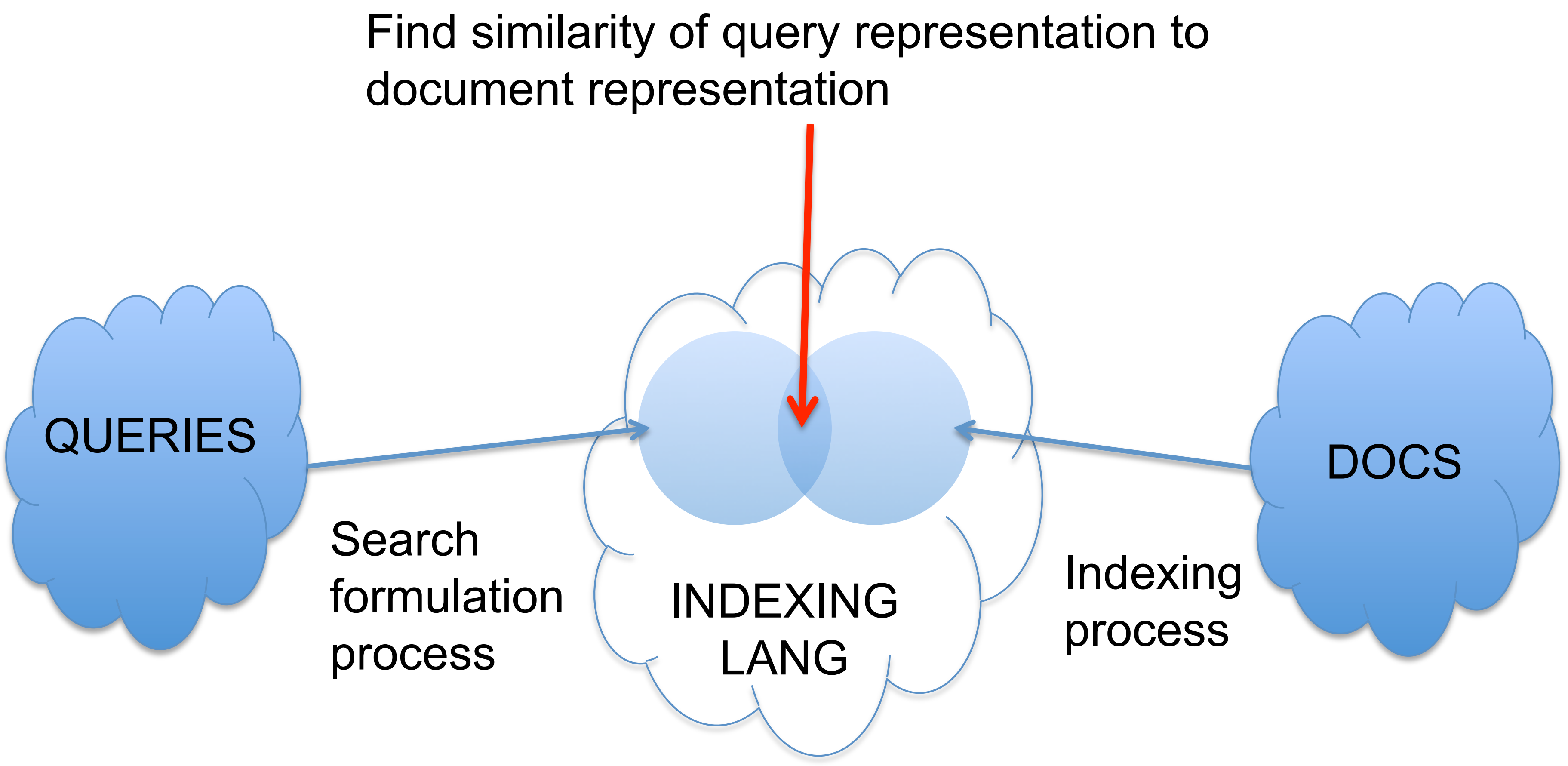
IR system basics

(Based on Salton & McGill, 1983)



But typically not a direct relationship

IR system basics


Find similarity of query representation to document representation
QUERIES
Search formulation process
INDEXING LANG
Indexing process
DOCS

Typically an indirect relationship

(Based on Salton & McGill, 1983)

Your task

- Groups 1..m (“search engines”):
  - Consider the document
  - Individuals:
    - Decide on 10 words you think best represent the document
    - Pick 1 word from these you consider the most important
- Groups m+1..n (“queries”):
  - Consider the information need
  - Individuals:
    - Decide on 10 words you think best represent the information need
    - Rank these in descending order of importance to need

Discussion: search engines

- How similar are lists of words you chose?
- Are any of your “most important” words the same?
- What is the difference between indexing a physical document and an electronic one?
- How did you choose your words?

Discussion: Queries

- How similar are your lists of words?
- Are any of your "most important" words the same?
- Are any of your other ranked words the same? (at the same rank…?)
- What if any background knowledge did you use?
- What about combining or structuring words?

Relevance

- Pair up: SE member with Query member
- Discuss:
  - Is document relevant to query? (binary judgement)
  - To what extent is it relevant? (graded judgement)
- Determine size of intersection if any between your SE list and your query list
  - **Would the query have found this document?**

MANCHESTER
1824
The University of Manchester

Information needs: Jansen et al. (2008)

- Identified 3 classes of need

1. Informational (~80% of queries)
   – Need to learn about something
2. Navigational (~10%)
   – Need to go to some page
3. Transactional (~10%)
   – Need to do something using the Web

Classify these into 1 of {Nav, Trans, Info}

- manchester weather
- easyjet
- blood pressure
- how to bake a cake
- beatles lyrics
- kenya
- bbc web page
- i’m feeling 40
- average temperature manchester weather
- gta 5
- gta 5 cheats
- yahoo.com
- 512*37
- mars rover images
- instanseation
- car hire dublin
- where was obama born
- nearest neighbour
- tesco tablet
- youtube
- hotels near heathrow
- tilt
- free school meals
- espresso coffee maker
- where is st lucia
- where is google
- navigate from york to bath

Identifying class

- How did you identify which class to put a query into?
- Navigational
  - – Company/business/org name
  - – Domain suffix
  - – Contains ‘web’
  - – Length of query (<3)

Identifying transactional

- Terms related to movies, songs, lyrics, recipes, images, humour
- Queries with ‘obtaining’ terms
- ‘download’ terms (inc. ‘software’)
- image, audio or video collections
- ‘audio’, ‘images’, ‘video’
- Entertainment terms (pictures, games, …)
- Interact terms (buy, chat,…)
- Movies, songs, images, multimedia, compression file extensions (jpg, zip, …)

MANCHESTER
1824
The University of Manchester

Identifying informational

- Use of question words
- Many phrases
- Informational items (‘list of…’)
- Query length >2
- Neither navigational nor transactional

  i.e., easier to identify N and T (?)

MANCHESTER
1824
The University of Manchester

Single or multi-class?

- How realistic is it to classify a query into just one class to guide a search?
- Revisit your classification
  - Assign 1..3 classes to each query, with weights (e.g. N:85%, T:10%)
  - Are there ‘unclassifiable’ queries?
  - How easy is it to figure out user intent from one query?

Users

- Make ill-defined queries, use imprecise words
- Make short queries
  - 2005: 2.54 words on average (KDD-Cup05)
  - 1998: 2.35 words (88% < 3 words) (Silverstein et al 98)
- Grammatical infelicities
- Spelling mistakes
- 78% of queries not modified (Google)
- 57% usually use the same search engine (iProspect survey 2004)

Google: User query spelling mistakes (importance/difficulty of spell checking)

488941	britney spears
40134	brittany spears
36315	brittney spears
24342	britany spears
7331	britny spears
6633	briteny spears
2696	britteny spears
1807	briney spears
1635	brittny spears
1479	brintey spears
1479	britanny spears
1338	britiny spears
1211	britnet spears

… (many, many more)

2	brynty spears
2	brythey spears
2	bryttney spears
2	btiany spears
2	btirtney spears
2	btitiney spears
2	btittny spears
2	btritany spears
2	buttney spears
2	grittney spears
2	prietny spears
2	pritany spears
2	prittany spears

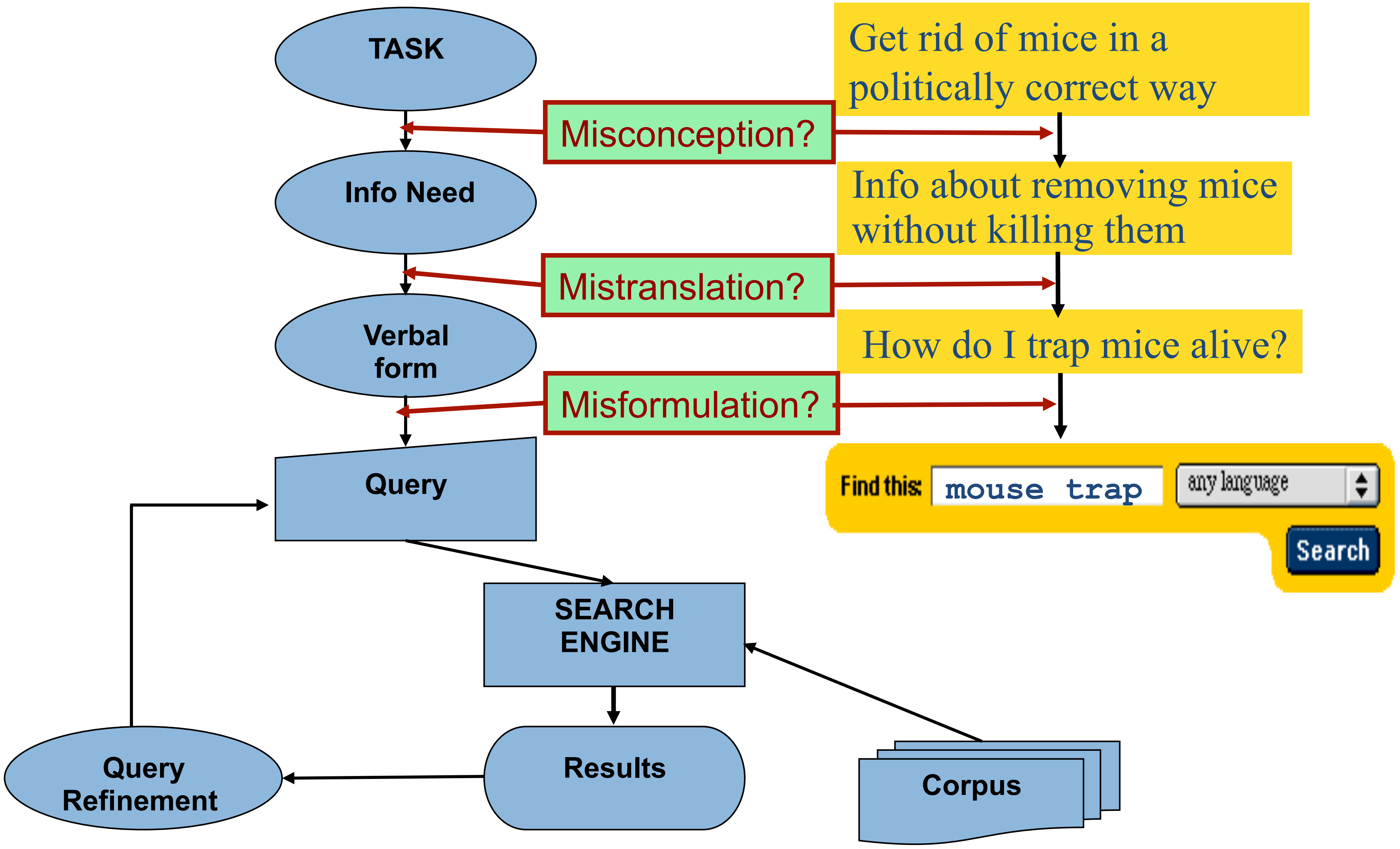
…

http://labs.google.com/britney.html (but Google Labs now closed...)

The University of Manchester

(Manning et al., IIR book)

The classic search model



Indexing to support search

- Documents must be indexed
- Index construction is critical
- Search is focused on index
- Index language: means of representing documents and of representing queries
- Map queries to index
- Determine similarity of query to document via index items

MANCHESTER 1824 The University of Manchester

Classic IR: reduction of information

- ‘Meaning’ of document reduced to set of index terms
- Highly reductive approach
- Problem becomes: match query terms with index terms associated with document
- Simplest model: “bag of words”
- NOTE: “index term” is NOT same as terminological term of a natural language

(IIR book)

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

- 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

(IIR book)

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.

MANCHESTER 1824

The University of Manchester

Massive matrix issues

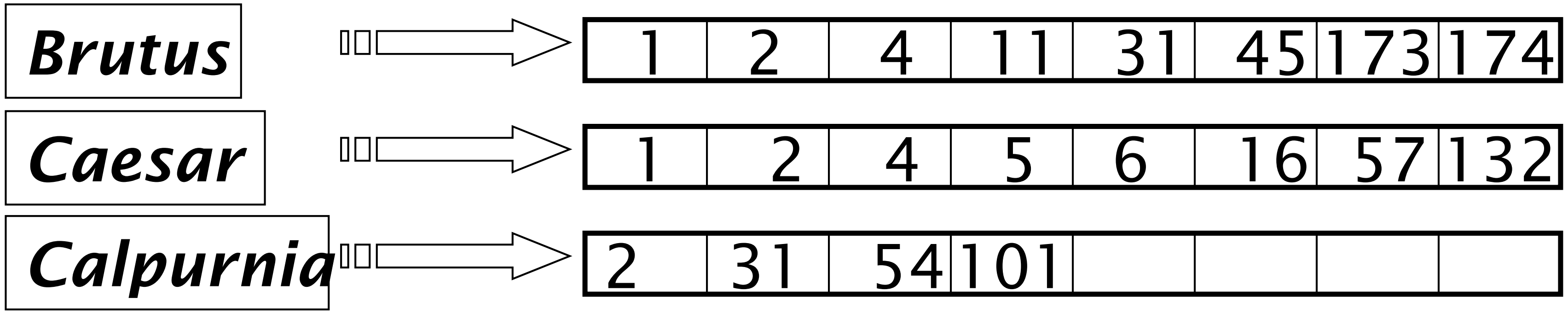
- Small collection has 1 million documents
  - With ~1000 words per document
- Assume 500000 unique words in collection
- Matrix: 500000 * 1000000 = 500000000000 (half a trillion) 1s and 0s
- Observe: sparse matrix! No more than 1 billion 1s

  → **Just store the 1s**

Inverted index

(IIR book)

- For each term *t*, we must store a list of all documents that contain *t*
  - Identify each by a **docID**, a document serial number



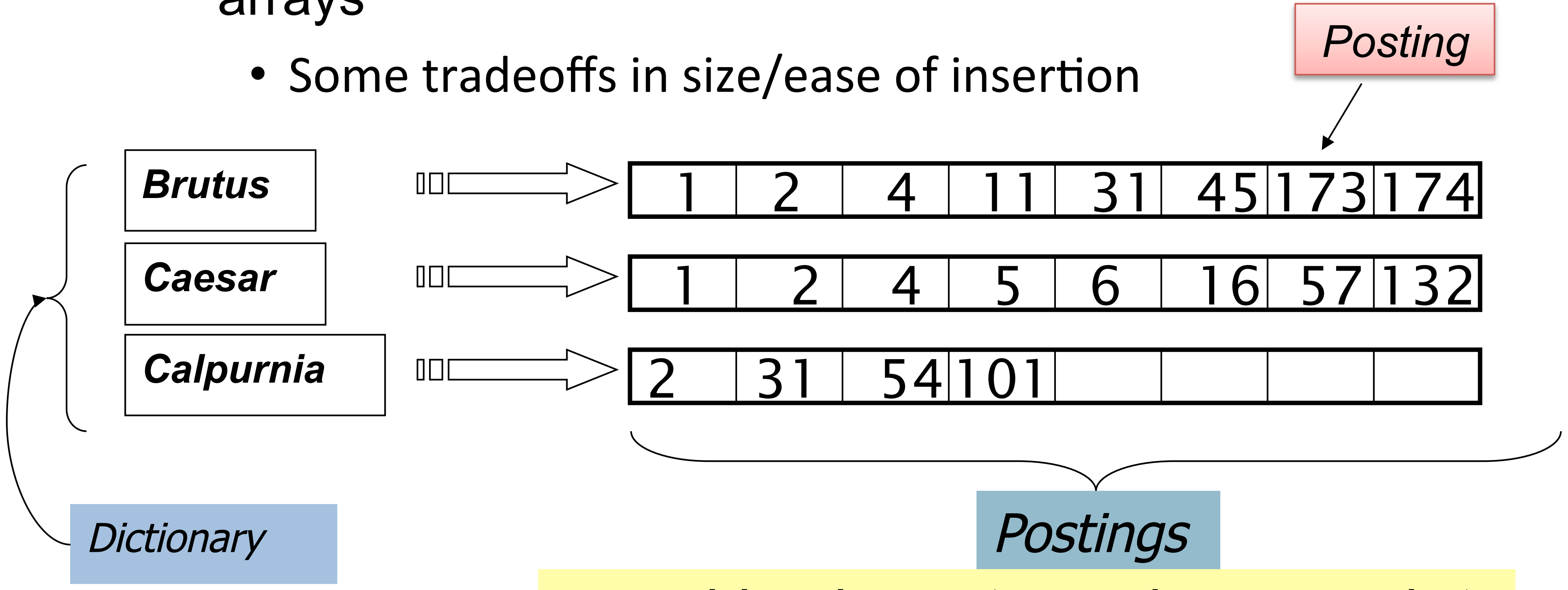
What happens if the word ***Caesar*** is added to document 14?

MANCHESTER 1824 The University of Manchester

Inverted index

(IIR book)

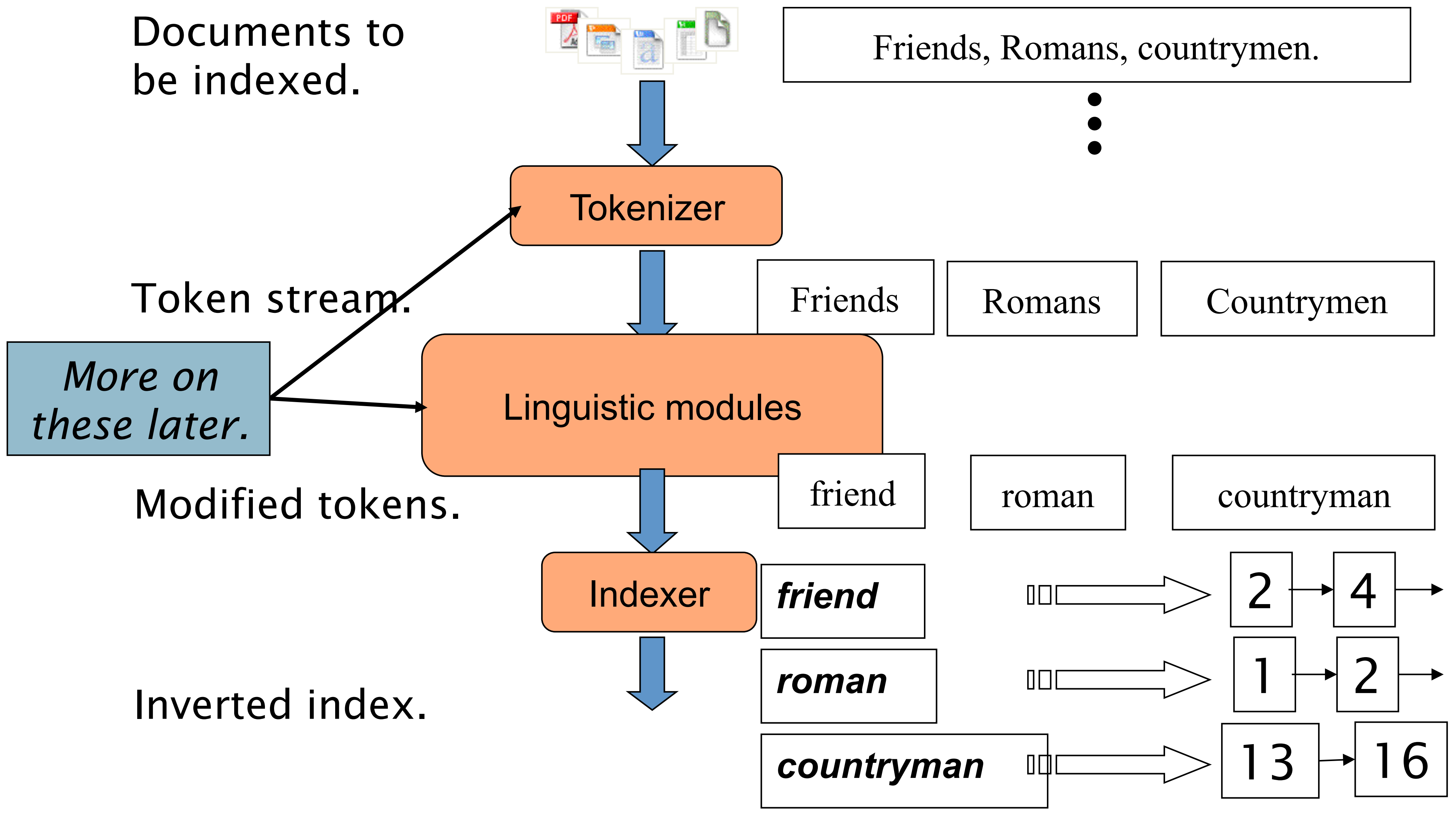
- We need variable-size postings lists
  - On disk, a continuous run of postings is normal and best
  - In memory, can use linked lists or variable length arrays
    - Some tradeoffs in size/ease of insertion



Sorted by docID (more later on why).

(IIR book)

Inverted index construction



MANCHESTER 1824
The University of Manchester

(IIR book)

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

(IIR book)

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

(IIR book)

Indexer steps: Dictionary & Postings

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

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

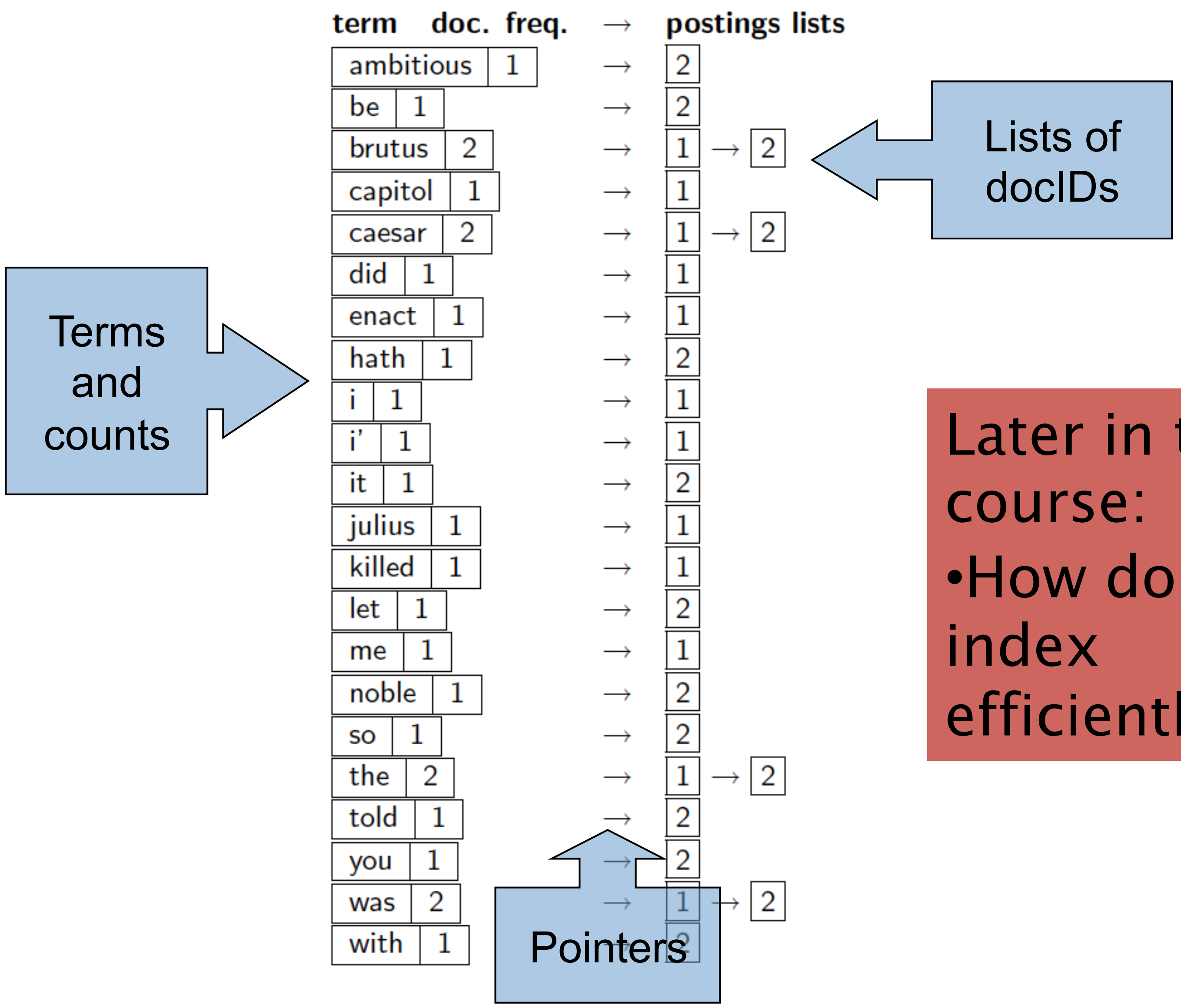
term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Why frequency?
Will discuss later.

Sec. 1.2

(IIR book)

Where do we pay in storage?



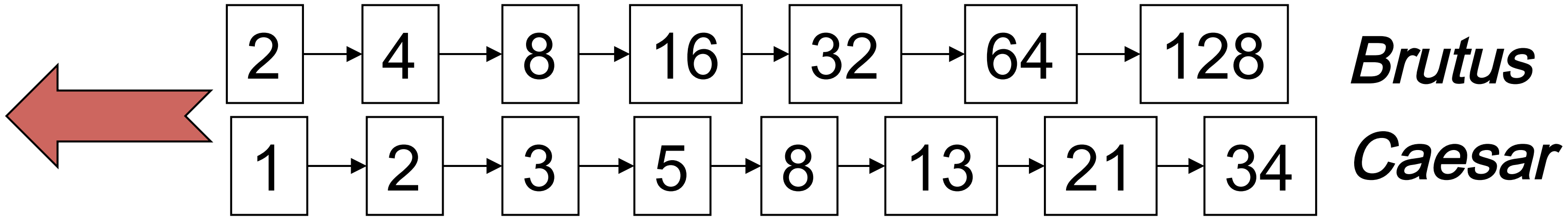
term	doc. freq.	postings lists
ambitious	1	2
be	1	2
brutus	2	1 → 2
capitol	1	1
caesar	2	1 → 2
did	1	1
enact	1	1
hath	1	2
i	1	1
i'	1	1
it	1	2
julius	1	1
killed	1	1
let	1	2
me	1	1
noble	1	2
so	1	2
the	2	1 → 2
told	1	2
you	1	2
was	2	1 → 2
with	1	2

Later in the course:
- How do we index efficiently?

Query processing: AND

- Consider processing the query:

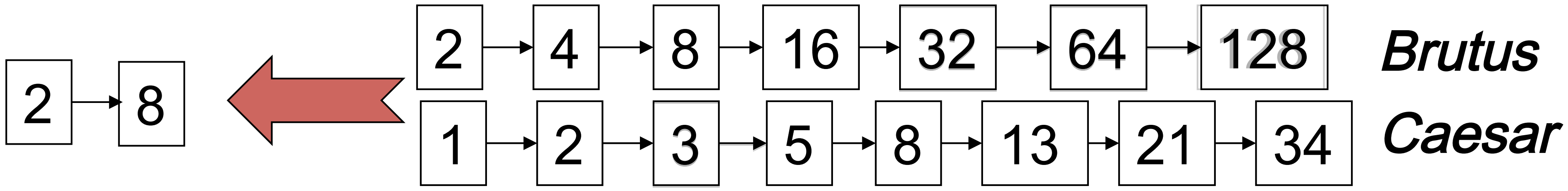
  Brutus** AND **Caesar
  - Locate ***Brutus*** in the Dictionary
    - Retrieve its postings
  - Locate *Caesar* in the Dictionary
    - Retrieve its postings
  - "Merge" the two postings:



(IIR book)

The merge

- 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.

Your turn

- Document collection:

Doc1: breakthrough drug for schizophrenia

Doc2: new schizophrenia drug

Doc3: new approach for treatment of schizophrenia

Doc4: new hopes for schizophrenia patients

1. Draw the term-document incidence matrix
2. Draw the inverted index for this collection

Boolean queries

- Use operators
  - AND (set intersection)
  - OR (set union)
  - NOT (set difference, complement)
- What is result set for:

1. schizophrenia AND drug
2. for AND NOT (drug OR approach)
3. new AND patient

Boolean queries

- What is time behaviour of:

1. Brutus AND NOT Caesar
2. Brutus OR NOT Caesar

- Is it still O(*x*+*y*) for both?

Arbitrary Boolean formulae?

(Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

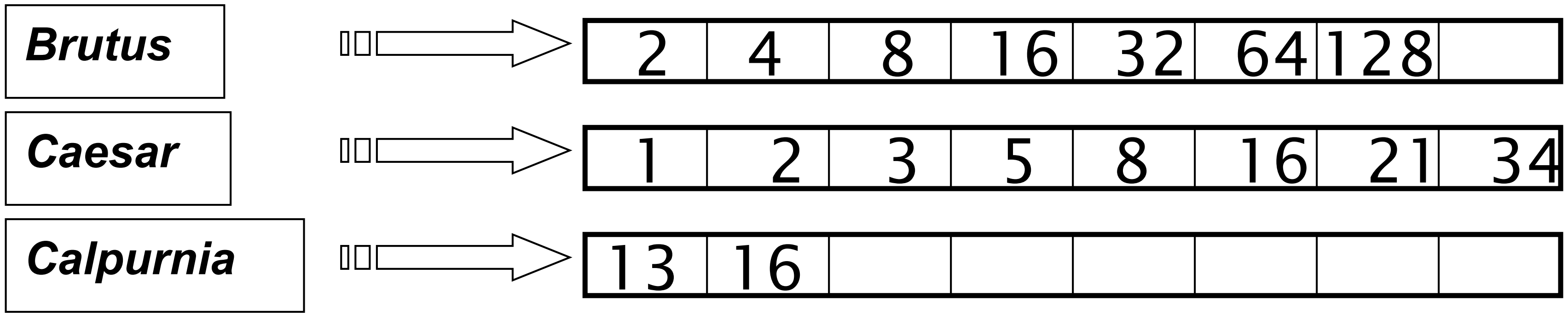
Can we always merge in linear time?

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



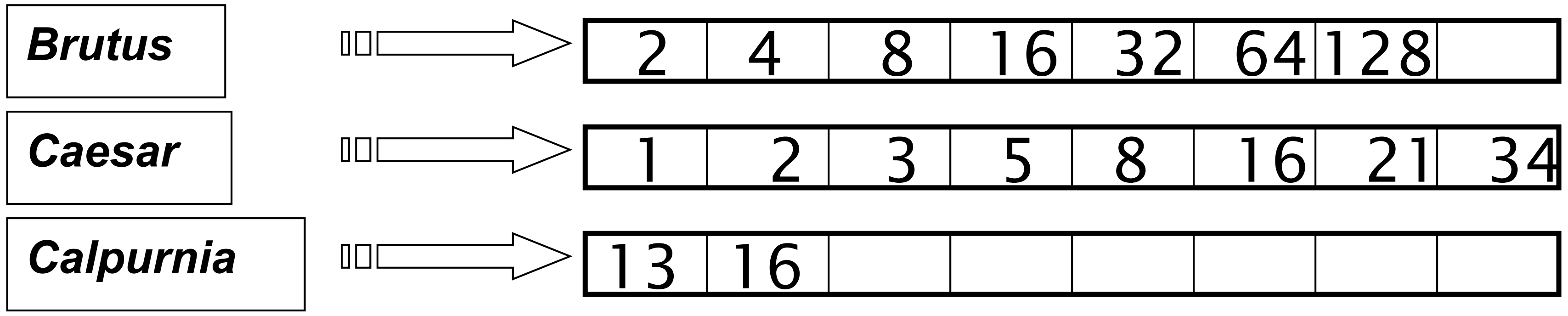
Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

(IIR book)

Query optimization example

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

This is why we kept document freq. in dictionary



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

MANCHESTER 1824
The University of Manchester

(IIR book)

More general optimization

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

The University of Manchester

(IIR book)

Recommend query processing order

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

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

MANCHESTER 1824
The University of Manchester

(IIR book)

Query processing exercise

- If query is ***friends*** *AND* ***romans*** *AND (NOT* ***countrymen****)*, how could we use the freq of ***countrymen***?
- For very frequent negated terms
  - Use ***N*-(length of postings list)**, not (length of postings list) in deciding order
- For infrequent negated terms
  - Use (length of postings list) for ordering. Process latter group last

Boolean model: pros

- Simple model, easy/efficient to implement
  - Document = set of words
- Precise
  - Document matches or not (inclusion/exclusion)
- Widely used for commercial, legal retrieval and for specialist searches
  - Long, precise queries
  - Works well when know what wanted
  - User feels in control
- Many search systems ‘hide’ Boolean functionality (e.g. space = AND)

Boolean model: cons

- “feast or famine”
  - AND gives too few or no results
    (the more ANDs the smaller the result set)
  - OR gives too many
- No relevance ordering/ranking of results (although may be in date reverse order)
- Order of words in documents not used
- Basic Boolean expressions too limiting for information needs

Boolean model: cons

- Non-expert user does not understand Boolean operators
- social AND worker AND union
  - Will not retrieve a document that has social and worker in it (with no union in it)
  - Documents that are ‘close’ are not retrieved
- Confusion with natural language
  - cats OR dogs : yields union
  - cats OR dogs AND NOT horses : any document with either cats or dogs or both will not be returned if it has horses in it

Extended Boolean model

- Basic Boolean model plus other operations
- Proximity operators
  - /n e.g. /3 – within 3 words
  - /s = in same sentence, /p in same para
  - +s = term1 must precede term2 in same sent
- Truncation: object! → object, objected, …
- Wildcard: reali*e → realise, realize

Extended Boolean model: Westlaw

- Largest commercial (subscription) legal search service
  - Available via UML: http://www.library.manchester.ac.uk/using-the-library/students/training-and-skills-support/my-learning-essentials/online-resources/
  - Under "searching" tab, find "Getting Results"
  - Choose Westlaw once into the tutorial
- Once connected to Westlaw: Help under "List of Connectors" link
- **NB: Space is "AND"** (used to be "OR")
- Try following (from IIR book with modification to allow for OR → AND)

Westlaw example search

- Information need: Information on legal theories involved in preventing disclosure of trade secrets by employees formerly employed by a competing company
- Query:

  "trade secret" /s disclos! /s prevent /s employe!
- When results appear, click on “show terms in context” link

Westlaw example searches

- IN: Requirements for disabled people to be able to access a workplace
- Query:

disab! /p access! /s (work-site OR work-place OR (employment /3 place))

- IN: Cases about a host's responsibility for drunk guests
- Query:

host! /p (responsib! OR liab!) /p (intoxicat! OR drunk!) /p guest

(Exercise)

- Group: using Westlaw, search for documents about boundary disputes between landowners/neighbours
- To do so, compose at least 5 queries using combinations of:
  - & (AND), or (OR), /p, /s, +s, +p, /n, +n
  - Where ‘n’ is a number
- You don’t have to use all in the same query!
- Note what differences you observe when using the + variants as opposed to the ones without +
- Show your results to one of us and be ready to report back to the class

MANCHESTER 1824
The University of Manchester

A real specialist search

- For a systematic review, searching several major online bibliographic databases
  - Tripney, J., Newman, M., Bird, K., Thomas, J., Kalra, N., Bangpan, M., Vigurs, C. (2010) *Understanding the drivers, impact and value of engagement in culture and sport: Technical report for the systematic review and database*. London: EPPI-Centre, Institute of Education, University of London.
- https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/88448/CASE-systematic-review-technical-report-July10.pdf
- Look at the queries in **Appendix 3, page 26ff**
- Explanation of these given on course Web site

Learning summary

- Documents can be represented in multiple ways
- An information need can be represented in many ways by a query
- There are many relevance judgements that can be made between a document and a query/need

Learning summary

- What and how you index is crucial
  - As classic IR is massively reductive
  - Indexing wordforms is not enough
- Simple Boolean model is insufficient for most users and confusing for them
  - But still widely used for expert search and still underpins much processing
- Fast, informative searching (at scale) requires sophisticated, large-scale indexing capabilities

MANCHESTER 1824
The University of Manchester

Resources (journals available via UML e-journals)

- Chapter 1 of Manning et al., Introduction to Information Retrieval (online, free)
- Jansen, B.J., Booth, D.L. & Spink, A. (2008) Determining the informational, navigational, and transactional intent of Web queries. *Information Processing and Management* 44: 1251-1266.
- More in-depth/theoretical for those interested:
  - Jansen, B.J. & Rieh, S.Y. (2010) in *JASIST* 61(8): 1517-1534.
  - Cole, C. (2011) in *JASIST* 62(7): 1216-1231.
  - Borlund, P. & Dreier, S. (2014) in *Information Processing and Management* 50(4): 493-507.
  - Lewandowski, D. (2015) in *JASIST* 66(9): 1763-1775.

MANCHESTER 1824

The University of Manchester

UML Resources

- http://www.library.manchester.ac.uk/using-the-library/students/training-and-skills-support/my-learning-essentials/online-resources/
- Under "searching" tab: "Planning ahead: making your search work" – introduction to basic Boolean searching and extended operators (online tutorial). See also at this level:
- "Shopping for information" online tutorial
- You may be interested in some of the bookable sessions, e.g. "successful searching"
- (check out other resources on writing, presentations, references – for your project!)