DD2476 Search Engines and Information Retrieval Systems

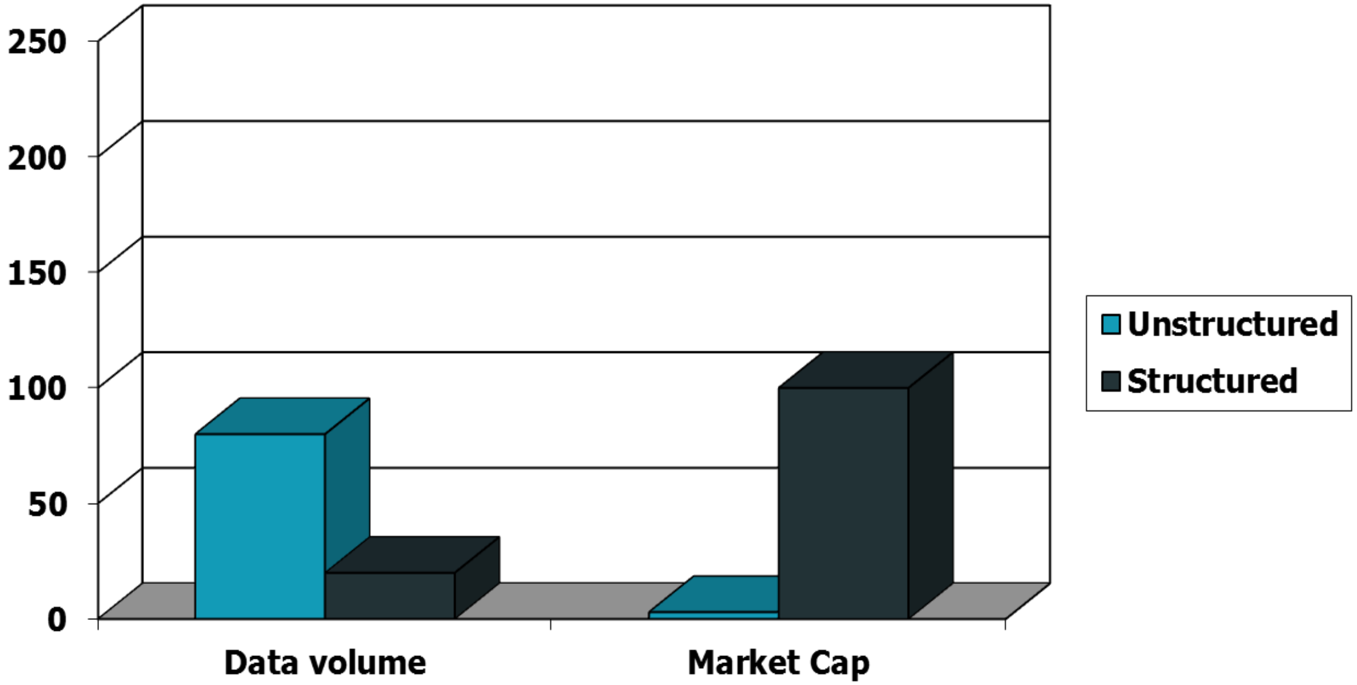
Lecture 1: Introduction

Hedvig Kjellström
hedvig@kth.se
https://www.kth.se/social/course/DD2476/

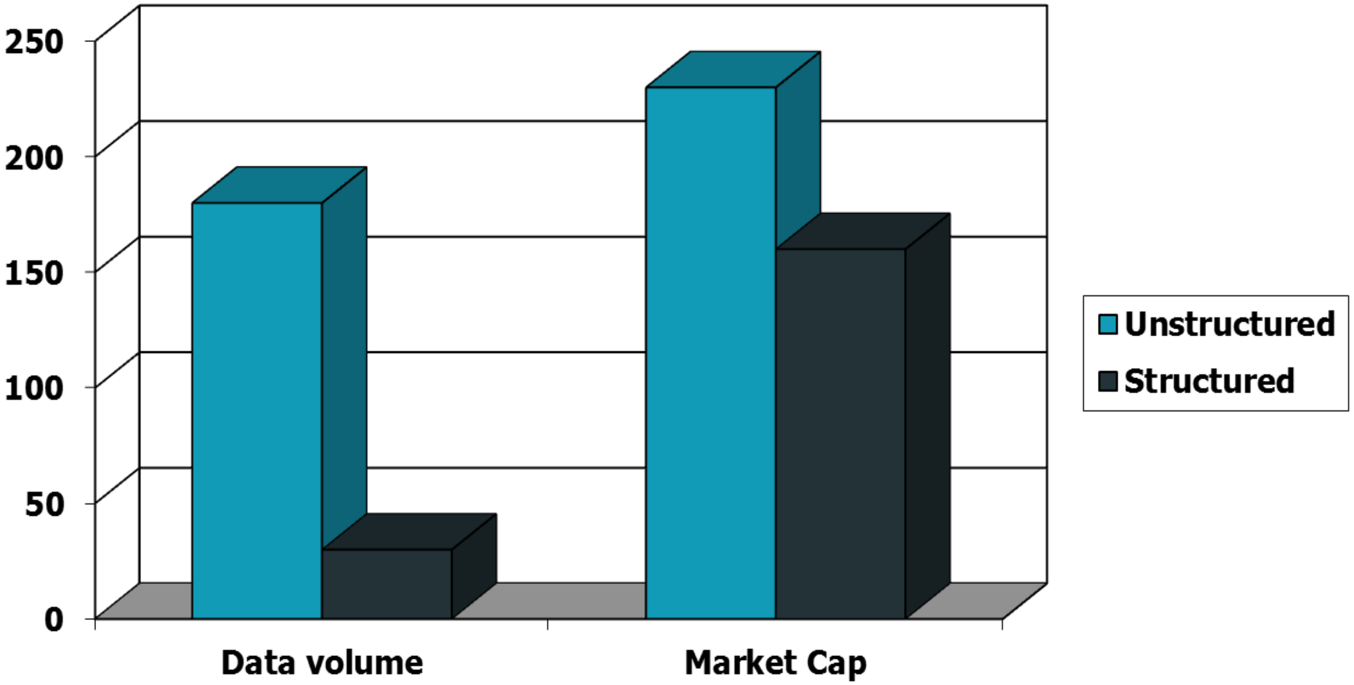
Definition

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 the mid-nineties



Unstructured (text) vs structured (database) data today



How good are the retrieved docs?

Precision: Fraction of retrieved docs that are relevant to the user’s information need

Recall: Fraction of relevant docs in collection that are retrieved

More in Lecture 3

Today

Presentation of lecturers

Course practicalities

- Curriculum
- Examination
- Course homepage: https://www.kth.se/social/course/DD2476

Boolean retrieval (Manning Chapter 1)

- Building an inverted index
- Boolean queries

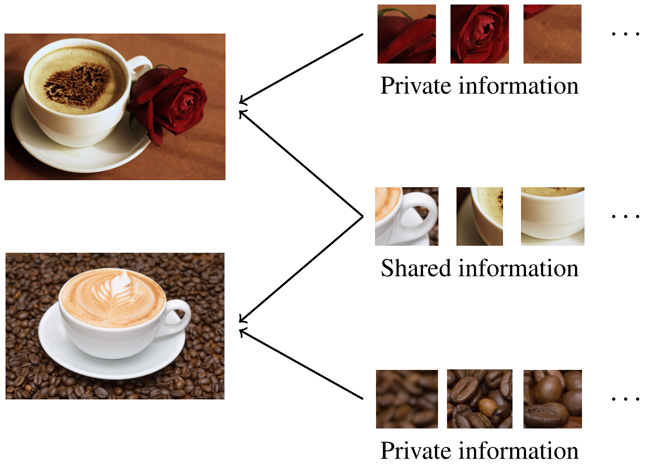
Presentation of Lecturers

Hedvig Kjellström

Associate Professor at CSC

Researcher in Robotics at CVAP, CSC

Lecture 1, 5, 6, 7



Zhang et al, CVPR subm 2016

Figure 1. An example of modeling ”a cup of coffee” images. Different images with a cup of coffee all share certain patterns, such as cup handles, cup brims, etc. Moreover, each image also contains patterns that are not immediately related to the ”cup of coffee” label, such as the rose or the coffee beans. They can be thought of as private for each image, or instance-specific.

Johan Boye

Associate Professor at CSC

Researcher in Language Technology at TMH, CSC

Lecture 2

Jussi Karlgren

Founder of Gavagai AB, Adjunct Professor at CSC

Researcher in Language Technology at TMH, CSC

Lecture 3, 4

Viggo Kann

Professor at CSC

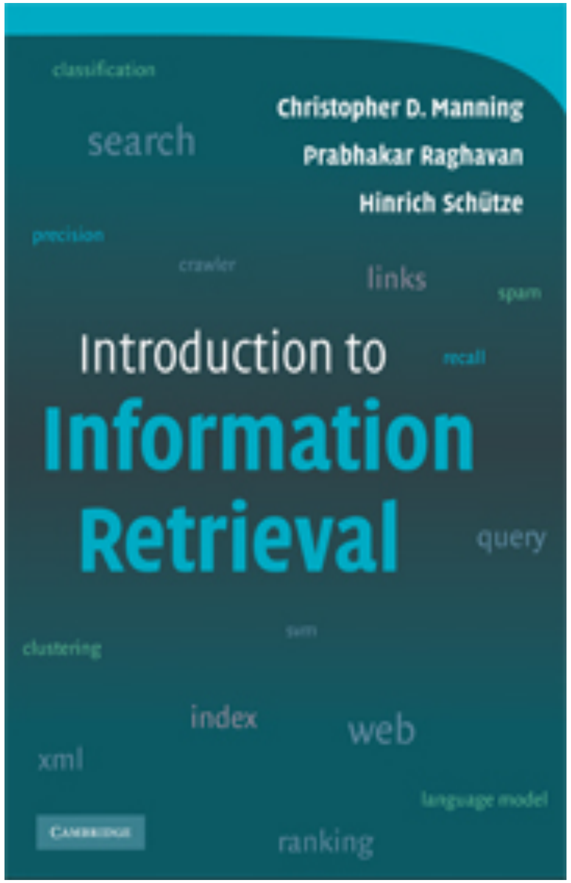
Researcher in Theoretical Computer Science at TCS, CSC

Lecture 8

Course Practicalities

Curriculum

C. D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008



Preliminary version available online in pdf format

- See course homepage: https://www.kth.se/social/course/DD2476

Curriculum

The major part of the book will be covered

Depth according to learning outcomes

- See course homepage: https://www.kth.se/social/course/DD2476

Reading on your own necessary

- Lectures cover only highlights, very high pace
- Examination on whole curriculum

Course given for the 7th time

- Changed the evaluation tasks in the assignments

Curriculum

Field moving forward at high pace

- This course a foundation to enable you to learn for yourself
- Source: Annual conference ACM SIGIR

Assignments: Give basics (**turn-of-the-century** search engine)

Project: Chance to have a glimpse of the state-of-the-art

Examination

Three computer assignments (6 ECTS, A-F)

- Individually
- Lab 1 (Lecture 1-3 readings) February 9
- Lab 2 (Lecture 4-6 readings) March 8
- Lab 3 (Lecture 7-8 readings) April 1

Project (3 ECTS, A-F)

- Groups of four-five students
- Presentation (Whole curriculum) May 20

Course Homepage

News!

Schedule with readings and examination deadlines

Contact information

Computer assignment and project descriptions

https://www.kth.se/social/course/DD2476

Set it to send you email!

Important

Boolean Retrieval

(Manning Chapter 1)

A First Information Retrieval Example

Ad hoc retrieval: Find documents in a collection of documents (corpus), relevant to a certain user need

Boolean retrieval model: Model in which queries are posed as Boolean expressions

Example: Shakespeare

- Find all Shakespeare plays that contain the words

BRUTUS AND CAESAR AND NOT CALPURNIA

BRUTE Force Approach

One could grep all of Shakespeare's plays for BRUTUS and CAESAR, then strip out plays containing CALPURNIA

- Unix command `grep`, linear search

Why is that not the answer?

- Slow (for large corpora)
- Other operations (e.g., find the word ROMANS NEAR COUNTRYMEN) not feasible
- Ranked retrieval (best documents to return)

Instead, organize beforehand

Term-Document Incidence Matrix

Document = play

Term = word

	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

Bitwise Operations

Document = play

Term = word

	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 AND NOT CALPURNIA

110100 AND 110111 AND NOT 010000

110100 AND 110111 AND 101111

= 100100 (**Antony and Cleopatra**, **Hamlet**)

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.

Exercise 5 Minutes

Consider 10^6 documents, each with ~10^3 words.

Avg 6 bytes/word including spaces/punctuation

- 6GB of data.

Say there are $0.5*10^6$ *distinct* terms among these.

Normal size collection!

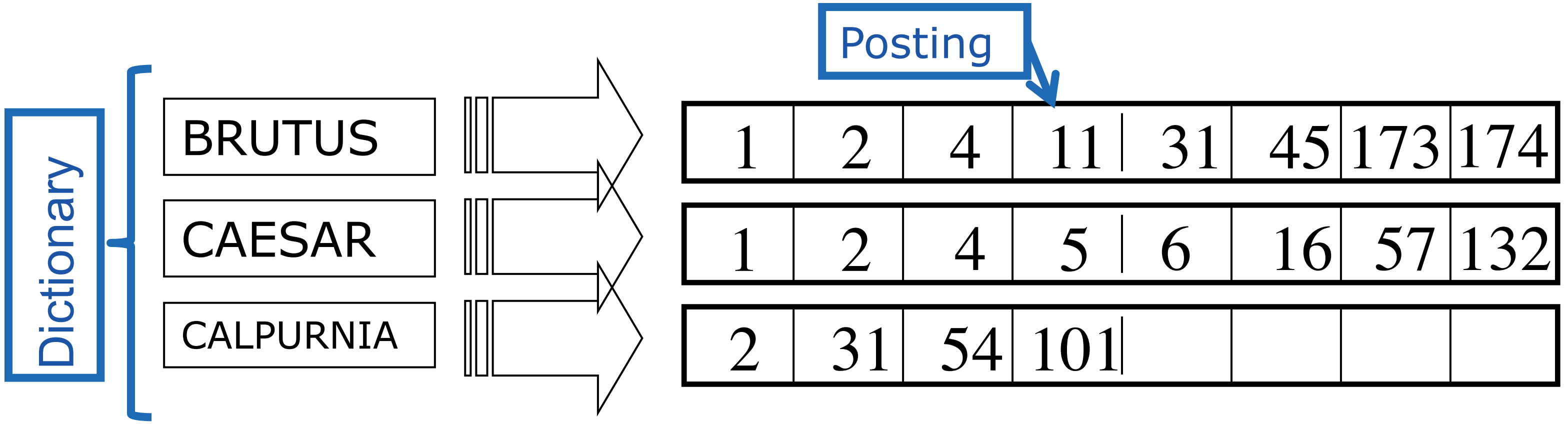
Discuss in pairs:

- What are the problems with using the term-document incidence matrix on a collection this size?
- How can the method be adapted to solve these problems?

Inverted Index

For each term *t*, store a list of all documents that contain *t*.

- Identify each by a docID, a document serial number



BRUTUS	1	2	4	11	31	45	173	174
CAESAR	1	2	4	5	6	16	57	132
CALPURNIA	2	31	54	101				

Can we use fixed-size arrays for this?

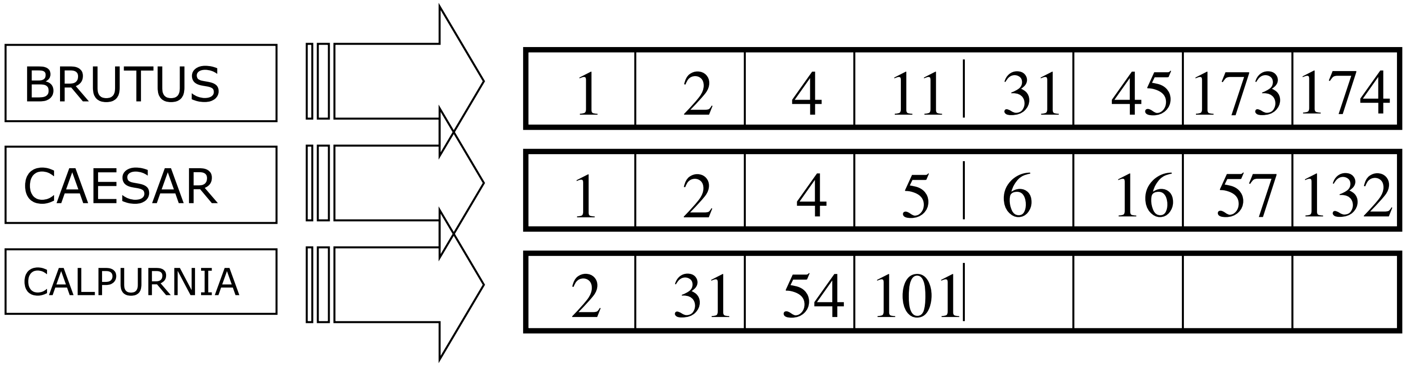
What happens if the term CAESAR is added to document 14?

Inverted Index

Need variable-size posting lists

Implementational details

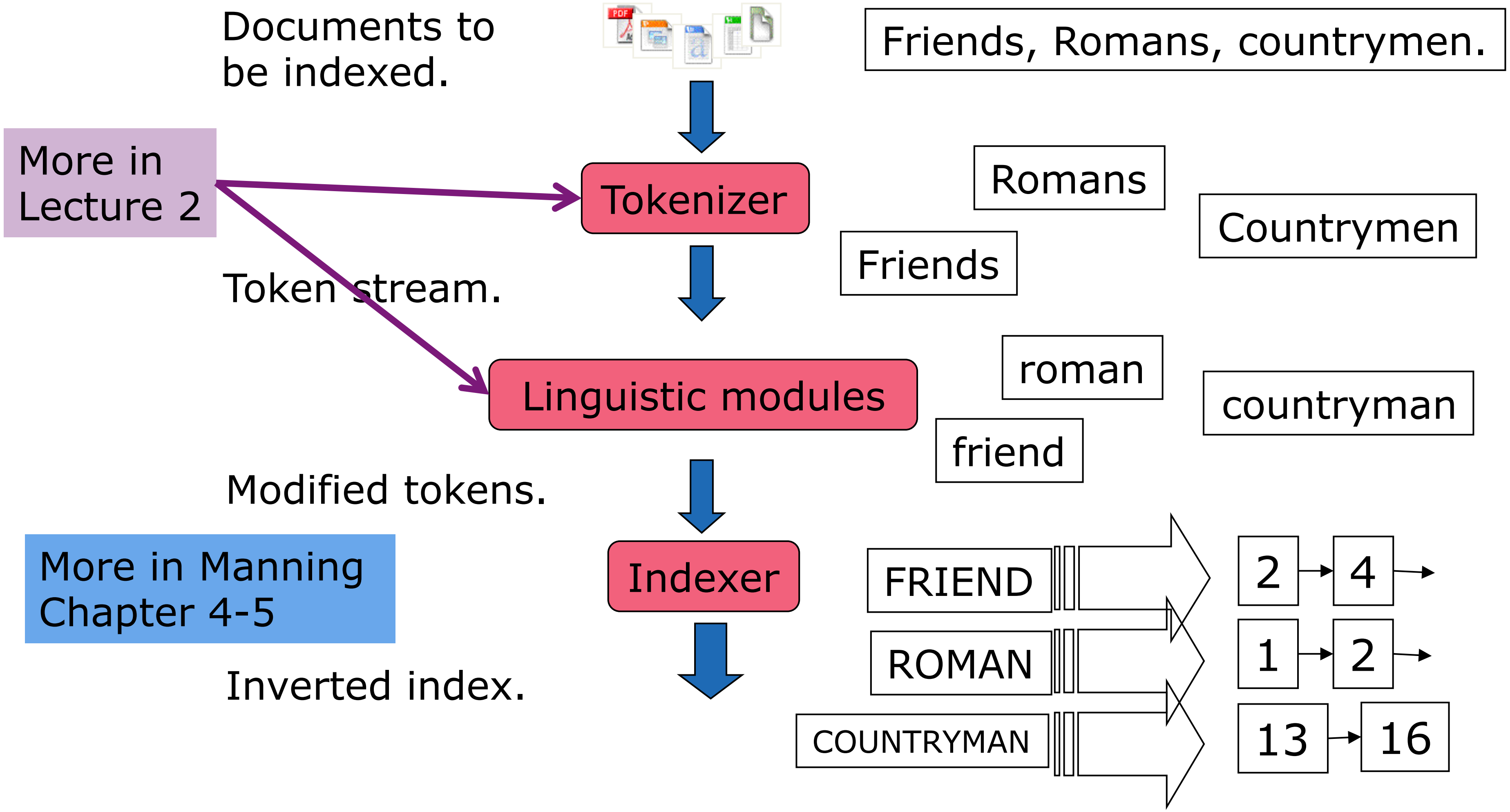
- trade-off storage size/ease of insertion
- Sort lists wrt DocID



BRUTUS	1	2	4	11	31	45	173	174
CAESAR	1	2	4	5	6	16	57	132
CALPURNIA	2	31	54	101				

More in Manning
Chapter 4-5

Building an Inverted Index



Documents to be indexed.

Friends, Romans, countrymen.

More in Lecture 2

Tokenizer

Token stream.

Friends Romans Countrymen

Linguistic modules

Modified tokens.

roman friend countryman

More in Manning Chapter 4-5

Indexer

Inverted index.

FRIEND 2 4

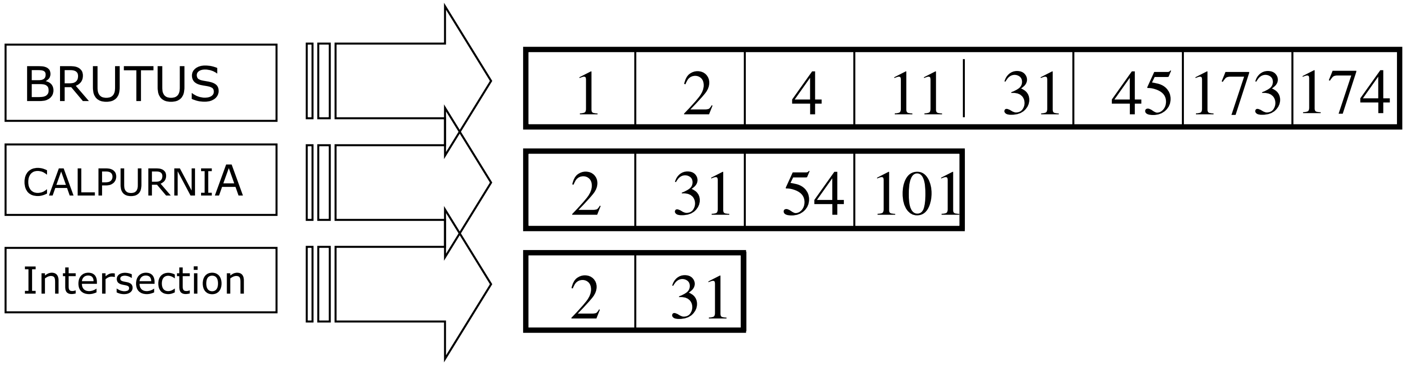
ROMAN 1 2

COUNTRYMAN 13 16

Query Processing with Inverted Index

Boolean queries are processed as with the incidence matrix

BRUTUS AND CALPURNIA



BRUTUS → 1 | 2 | 4 | 11 | 31 | 45 | 173 | 174

CALPURNIA → 2 | 31 | 54 | 101

Intersection → 2 | 31

NOT can also be handled with search

Organizing this work (sorting, evaluation order): query optimization

More in Manning Chapter 1

Beyond Term Search

Allow compounds, e.g., phrases ”…”

- ”FRIENDS, ROMANS, COUNTRYMEN!”

More in Lecture 2

Additional operators, e.g., NEAR

- CAESAR NEAR CALPURNIA
- Index has to capture term proximity

Zones in documents

- (author = SHAKESPEARE) AND (text contains WORSER)

More in Manning Chapter 10

Beyond Term Search

Not only presence/absence, but also term frequency

- 0 vs 1 hit
- 1 vs 2 hits
- 2 vs 3 hits
- Usually, more is better

More in Lecture 5

Exercise 5 Minutes

Try the search feature at www.rhymezone.com/shakespeare

- Who has an open browser? Find someone nearby, or come up to me.

Discuss in groups:

- What could it do better?
- Write down

IR vs Databases: Structured vs Unstructured Data

Employee	Manager	Salary
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

Typically allows numerical range and exact match (for text) queries, e.g.,

Salary ≥ 60000 AND *Manager* = Smith.

Unstructured Data

More in Lectures 9-12

Typically refers to free text but could also be

- Images
- Other media files

More in Lecture 5

Allows

- Keyword queries
- Free text queries e.g., find all web pages dealing with "drug abuse"
- Classic model for searching text documents

More in Lecture 7

No data is truly unstructured

- Grammar
- Semistructured search, e.g., XML

More in Manning Chapter 10

Organizing Data

Boolean queries only give inclusion or exclusion of docs.

More in Manning Chapter 16-17

Clustering: Given a set of docs, group them into clusters based on their contents.

More in Manning Chapter 13-14

Classification: Given a set of topics, plus a new doc *D*, decide which topic(s) *D* belongs to.

More in Lecture 5

Ranking: Can we learn how to best order a set of documents, e.g., a set of search results

The Web and Its Challenges

Unusual and diverse documents

Unusual and diverse users, queries, information needs

Beyond terms, exploit ideas from social networks

- E.g. link analysis

More in Lecture 6

How do search engines work? And how can we make them better?

More in Lectures 6, 9-12

Next

Lecture 2 (January 22, 10.15-12.00)

- D3
- Readings: Manning Chapter 2, 3

Computer Assignment 1 (now – February 9)

- Assignment description: https://www.kth.se/social/course/DD2476