Information Retrieval Methods

Helena Ahonen-Myka
Spring 2007, part 1
Introduction
Translation from Finnish: Greger Lindén

1. In this part...

- A definition
- Course administration
- An introduction to the field
- Course contents

2

Definition

- Information retrieval; information storage and retrieval (IR)
- Tiedonhaku; informationssökning
- ”IR is concerned with the processes involved in the representation, storage, searching and finding of information which is relevant to a requirement for information desired by a human user.” (Ingwersen, 1992)

3

Course administration

- Master level, 6 ECTS
- lectures (Helena Ahonen-Myka)
  - 15 January - 20 February, 2007: Mon, Tue 10-12 B222
- exercises (Niina Haiminen)
  - 22 January - 19 February, 2007, Mon 12-14, C221

4

Course administration

- project work
  - Project definition on web page
  - Group work: 4-5 students in each group formed during the first exercise session
  - The group returns a report about the work
  - The group presents their work during the last exercise session
- exam: Mon 26 February 9-12
- lecture notes
  - Slides on course web page
  - References given for each lecture; see course web pages

5

Course administration

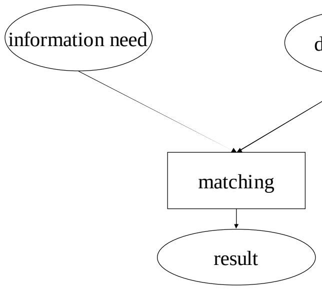
- Attending lectures is optional
- Attending exercise sessions is optional
  - but you can get max 5 points for tasks that you solve in advance
  - tasks will be available on the course web page on (previous) Monday
- Project work is obligatory
  - gives max 15 points
- Exam is obligatory
  - gives max 40 points

6

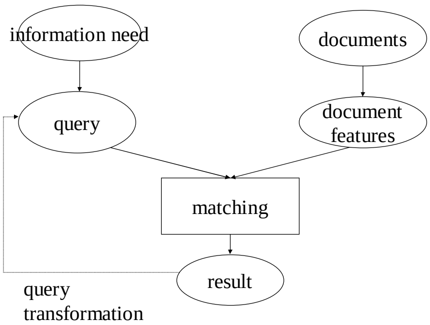
Introduction

- The goal of information retrieval is to satisfy needs for information
- Information retrieval strives to find the document or document set that satisfies the information needs in the best possible way

The information retrieval process



The information retrieval process



Different information needs

- Searching for a topic
  – "climate change"
- Individual search
  – "The PhD thesis of Esko Ukkonen"
- Searching for facts
  – "The Chancellor of the University of Helsinki in 2002"

Different kinds of documents

- Strictly structured (relational database)
  – E.g. library database, time tables,…
- Semi-structured
  – E.g.. reference database: meta data for the publications and their abstracts
  – XML documents
- Unstructured text documents
  – But they do contain some textual structure

Information needs vs. different kinds of documents

- queries can be exact or approximate
  – Exact query: relevant documents can be described with some features in an unambiguous way
  – Approximate query: relevant documents cannot be described with some features or in an unambiguous way

Information needs vs. different kinds of documents

- Exact database query
  - “students that major in computer science and started their studies in 2001”
    - attributes: first year, discipline
  - The answer is always correct (unless the database contains errors)
- A database query can also be approximate
  - The system could return students that started their studies in 2000-2002 (e.g. if there were no students starting in 2001)

13

Information needs vs. different kinds of documents

- queries on full text are usually approximate: e.g. ”climate change”
  - It is hard to know which terms have been used in different documents that discuss this topic: “climate change global warming weather carbon emission… “
  - many other topics may have been described with the same terms
  - → the result is often incomplete
  - → the result may contain irrelevant documents

14

Information needs vs. different kinds of documents

- queries on semi-structured documents combine exact and approximate queries
  - ”books written by John Irving containing a character named Jack”

15

Information needs vs. different kinds of documents

- the result of a query may be direct or indirect
  - Direct: the answer is found in the result
  - Indirect: the result contains pointers to the sources of the information that was searched for, e.g., literature references to documents or addresses of companies

16

Information needs vs. different kinds of documents

- Previously, information retrieval systems (text databases) and database management systems (ordinary databases) were two different things (in research or product development)
- Today we have integrated systems that can manage both structured and unstructured information (including pictures and multimedia)
  - Compare with XML: there is no border between strictly structured and unstructured text

17

Representations of queries and documents

- matching unstructured, natural-language queries and documents is difficult
- → both queries and documents must be represented in a more suitable way
  - Often by a set of terms
  - term = a unit of semantic expression, e.g.. word, phrase, stem (of a word)

18

Representations of queries

- A set of index terms (key words)
- An expression, where index terms have been combined with Boolean operators
  - "John and Irving"
  - "(text or image) and retrieval"
- Terms combined with proximity (nearness) operators
  - "John near Irving"

19

Representations of queries

- also sentences in natural language
  - E.g. a question-answering system accepts questions as input
    - "Who was the Chancellor of the University in 2002?"
  - Also a retrieved document can act as a query
    - The system looks for documents that are similar
  - Sentences are preprocessed: stemming, removing too frequent words (stopwords)
  - The system may also perform deeper analysis: what is the question word, what is expected as an answer

20

Representation of documents

- A document can be represented
  - Automatically based on terms that have been selected from the document on statistical grounds
  - Automatically based on terms that have been selected from the document on linguistic grounds
  - With terms selected by a human
- These alternatives can be combined

21

Representing a document collection

- a fast answer is usually required
- a document set can be very large
- there can be a very large number of terms (~ 10 000 – 100 000)

- → it is not possible (=it is inefficient) to use string search that scans the whole text (e.g.. UNIX grep)
- → it is not possible (=it is inefficient) to build separate indices for each term or combination of terms (compare with database indices for different attributes)

22

Representing a document collection

- Instead we construct an inverted index (inverted file) which helps to find occurrences of any term in the document collection efficiently
- Indexing = selection of terms + construction of an inverted index
  - an operation that takes time and is not performed very often

23

A document collection and its inverted index

The documents D1-D4 contain terms T1-T4 (0 = no, 1 = yes)

	T1	T2	T3	T4
D1	1	1	0	1
D2	0	1	1	1
D3	1	0	1	1
D4	0	0	1	1

The inverted index is

	D1	D2	D3	D4	
T1	1	0	1	0	"inverted list"
T2	1	1	0	0	
T3	0	1	1	1	
T4	1	1	1	1	

24

Dictionary file

- A dictionary file is the list of indexed terms (selected to the inverted index)
  - It also specifies the frequency of the term and gives a link to the inverted index

T1	2	→
T2	2	→
T3	3	→
T4	4	→

25

Matching a query to a document

- If there is only a single word in the query
  - The system looks for the term in the dictionary file
  - And with the help of the link, the correct position in the inverted index
  - And returns the set of documents in the inverted list
- If there are several words in the query, the inverted lists have to be merged
  - If the inverted lists of each term is represented in order according to the document number, two lists can be merged with one sweep

26

Processing Boolean queries

- A or B
  - the union of the inverted lists of A and B
- A and B
  - the intersection of the inverted lists of A and B
- A not B
  - the difference of the inverted lists of A and B
  - usually it is not possible to use the ‘not’ operator on its own: the result would be too large

27

Query: ((T1 or T2) and T3)

- from the inverted index:
  - T1: <D1, D3>
  - T2: <D1, D2>
  - T3: <D2, D3, D4>
- T1 or T2: <D1, D2, D3>
- (T1 or T2) and T3: <D2, D3>

28

Query: T1 and T3 and T4

- the query can be optimised based on the frequencies of the terms (that we find directly in the dictionary file)
  - T1: 2 documents <D1, D2>
  - T3: 3 documents <D2, D3, D4>
  - T4: 4 documents <D1, D2, D3, D4>
- T1 and T3: <D2>
- (T1 and T3) and T4: <D2>
- what happens if we first compute T3 and T4?

29

The proximity operator: A near B

- we may vary the indexing granularity
  - In addition to the document number, we could also specify in the inverted file the paragraph number, the sentence number, the exact position of the word
- we could find a result for the query ”A near B” directly with the help of the index
  - but the index would require a lot of memory
- If we have not stored position information that is precise enough in the index, we could first look for ”A and B” and then check the nearness by scanning the result documents

30

Searching for a string pattern through sequential search

- Basic problem: look for the occurrences of the pattern P in the string T
  - find all the occurrences of the word 'relevance' in a document
- Like the 'find' operation in the editor or the 'grep' command in Unix
- Use: more exact retrieval from the retrieval results with postprocessing
  - e.g. implementation of the proximity operator
- Can be used as a primary search method, if the document collection can fit into main memory

31

Course contents

- Relevance (the concept of), evaluation of information retrieval systems
- Indexing document collections
- Processing queries, matching methods
- Document clustering, information filtering and routing, other applications
- String matching by sequential search
- Processing natural language for information retrieval, multilingual information retrieval
- Other topics if time allows

32