Information Retrieval (IR)

Esther Pacitti

Partage de Données a Grande Echelle

Master 2 Informatique

Scenario

Information Need

- Example of an information need in the context of the world wide web:
- Find all documents (information!) about universities in India that
  - (1) offer schools of computer science and management degrees and
  - (2) offers data mining lectures. The information (the document!) should include full curriculum, fees, student campus, e-mail and other contact details.
- Formal representation of an information need = Query

Informal Definition

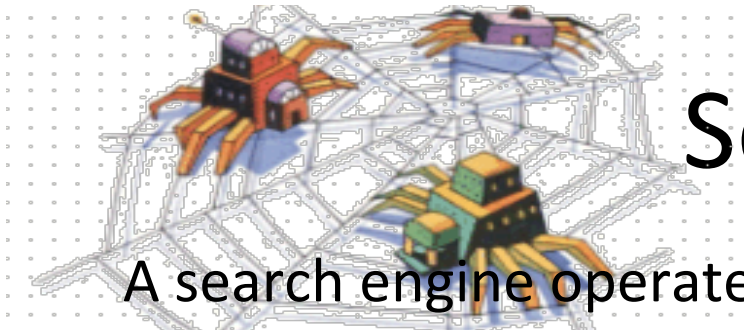
- Representation, storage, organisation and access of information (information items, information objects, documents). Not tuples.
- Find relevant (useful) information

Web Search Engine

Internet search engines are special sites on the Web that are designed to help people find information stored on other sites. There are differences in the ways various search engines work, but they all perform three basic tasks:

1. They search the Internet or select pieces from the Internet based on important keywords.
2. They keep an index of the words they find and where they find them.
3. They allow users to look for words or combinations of words found in that index.


WORLD
WIDE
WEB



Search Engine

A search engine operates, in the following order

Crawling

Follow links to find information

Indexing

Record what words appear where

Ranking

What information is a good match to a user query? What information is inherently good?

Displaying

Find a good format for the information

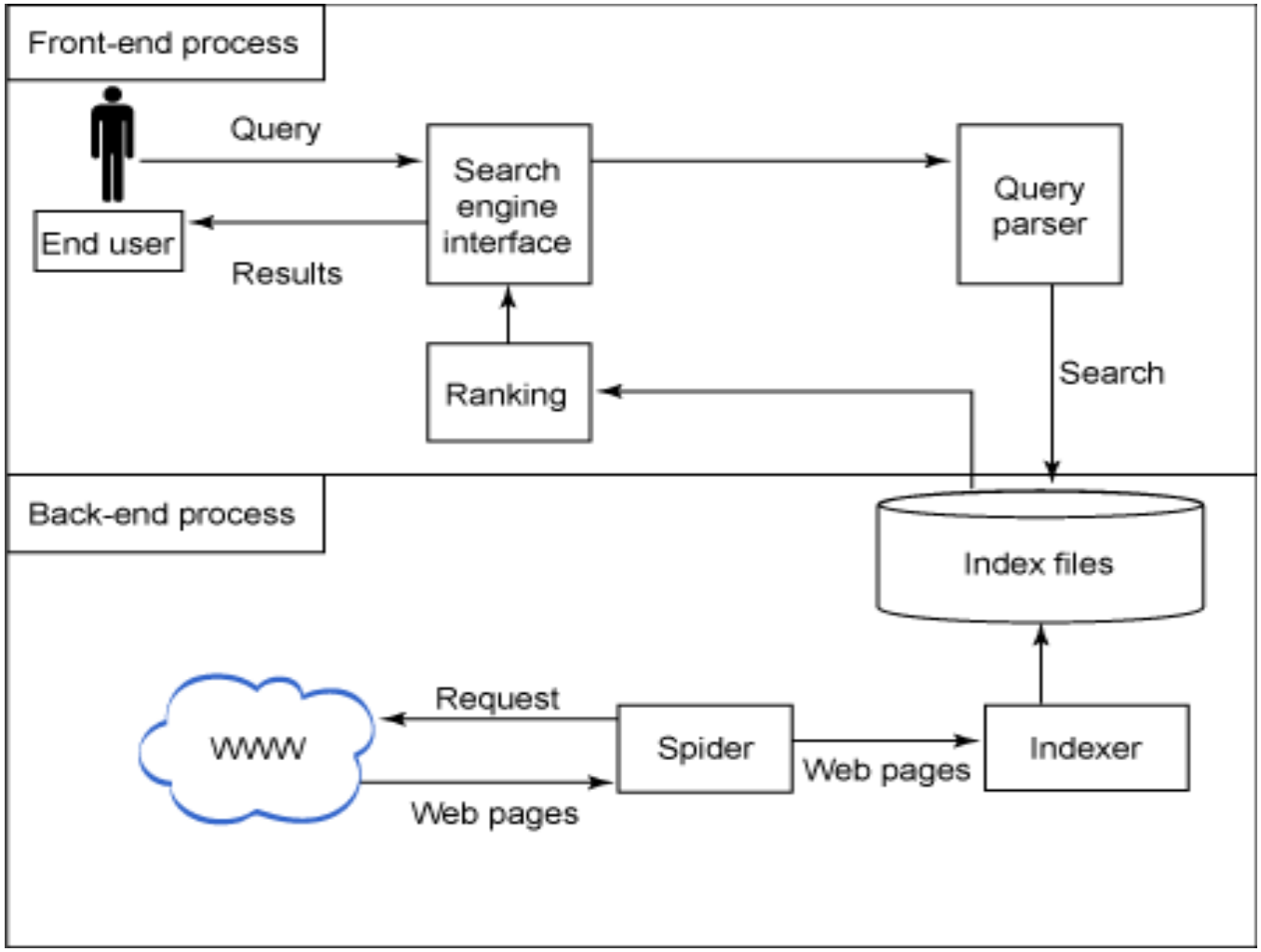
Serving

Handle queries, find pages, display results

Definitions

1. **Spiders:** To find information on the hundreds of millions of Web pages that exist, a search engine employs special software robots, called **spiders**, to build lists of the words found on Web sites.
   "Spiders" take a Web page's content and create key search words that enable online users to find pages they're looking for.
2. **Crawling**: When a spider is building its lists, the process is called **Web.** In order to build and maintain a useful list of words, a search engine's spiders have to look at a lot of pages.
3. **Indexing**: For fast accessing of data.
4. **Meta tag:** The contents of each page are then analyzed to determine how it should be **indexed** (for example, words are extracted from the titles, headings, or special fields).

Web Search Engine



Informal Definition

- Goal of an IR system - RECALL
  - Retrieve all relevant documents.
- Goal of an IR system - PRECISION
  - Retrieve the most relevant documents.
- Goal of an IR system:
  - Retrieve as few non-relevant documents as possible.
  - Retrieve relevant documents before non-relevant documents.

Recall and Precision

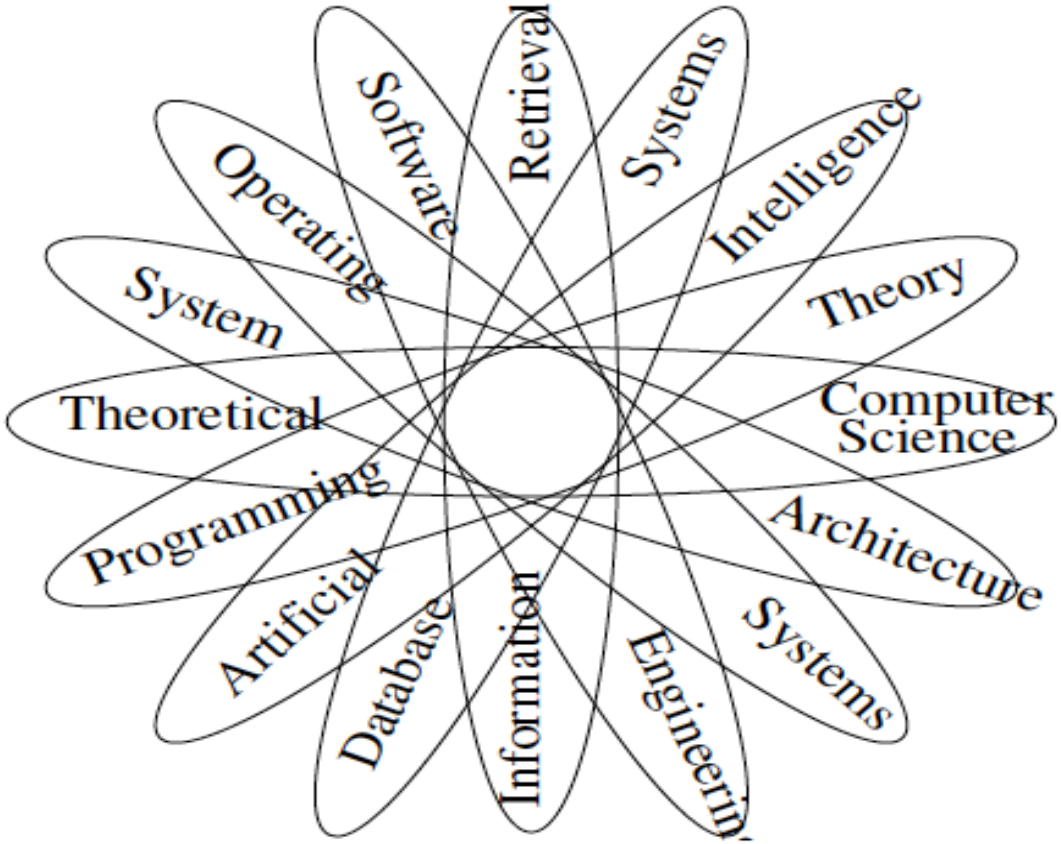
- Precision is the fraction of the documents retrieved that are relevant ti the user's information need.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

- Recall is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

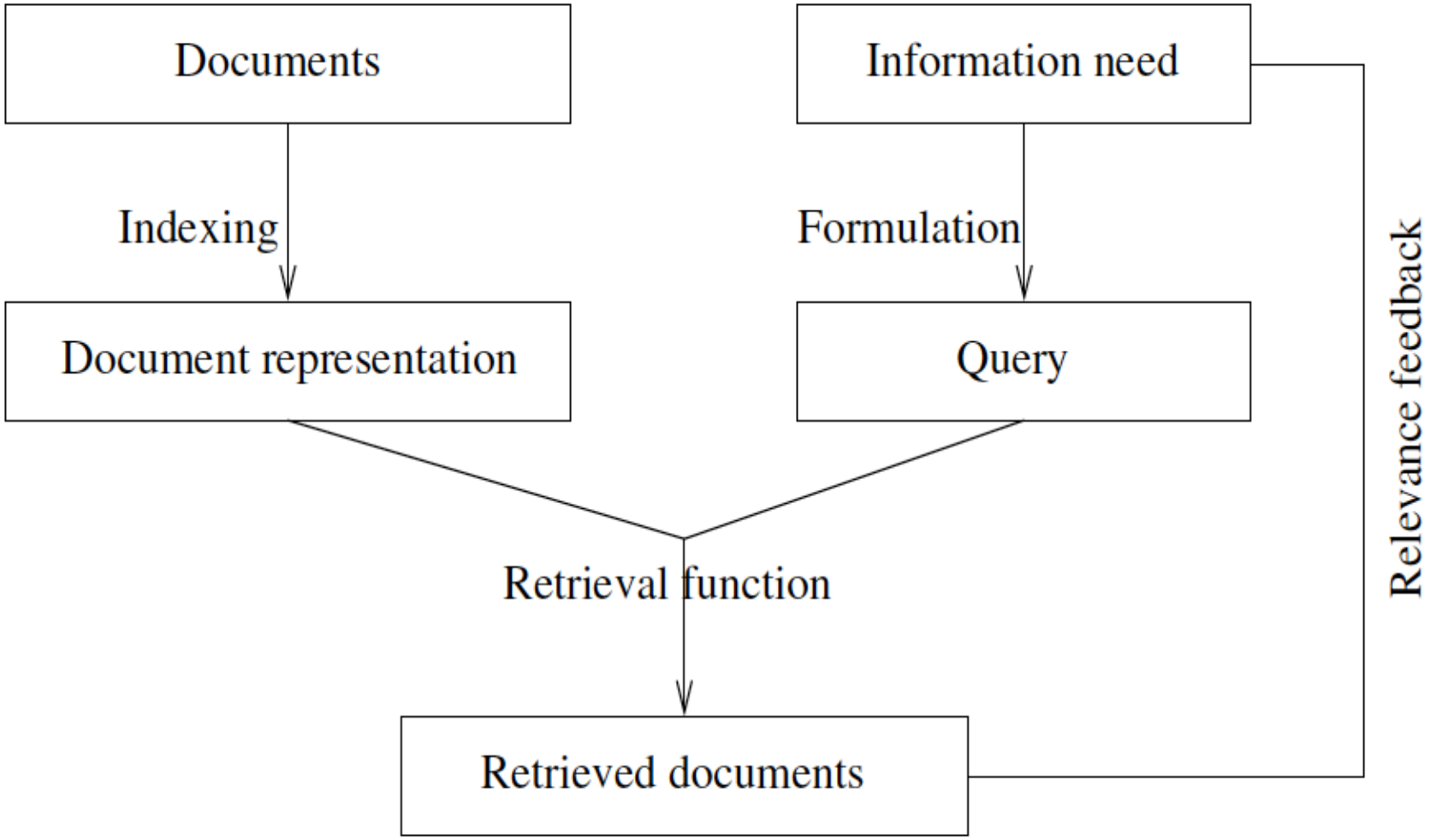
Information Retrieval in Computer Science



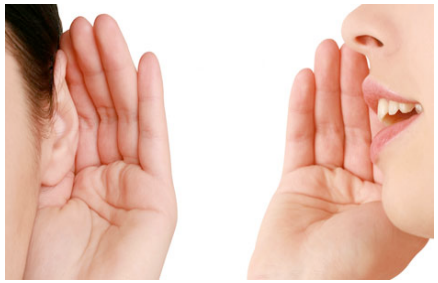
Information Retrieval x Data Retrieval

	Information Retrieval	Data Retrieval
Matching	vague	exact
Model	probabilistic	deterministic
Query language	natural	artificial
Query specification	incomplete	complete
Items wanted	relevant	all (matching)
Error handling	insensitive	sensitive

Model


Documents
Information need
Indexing
Formulation
Document representation
Query
Retrieval function
Retrieved documents
Relevance feedback

Information Deluge – BIG DATA



Natural Language

News

E-mails

Movies

Web Sites Pages

Photos

Documents

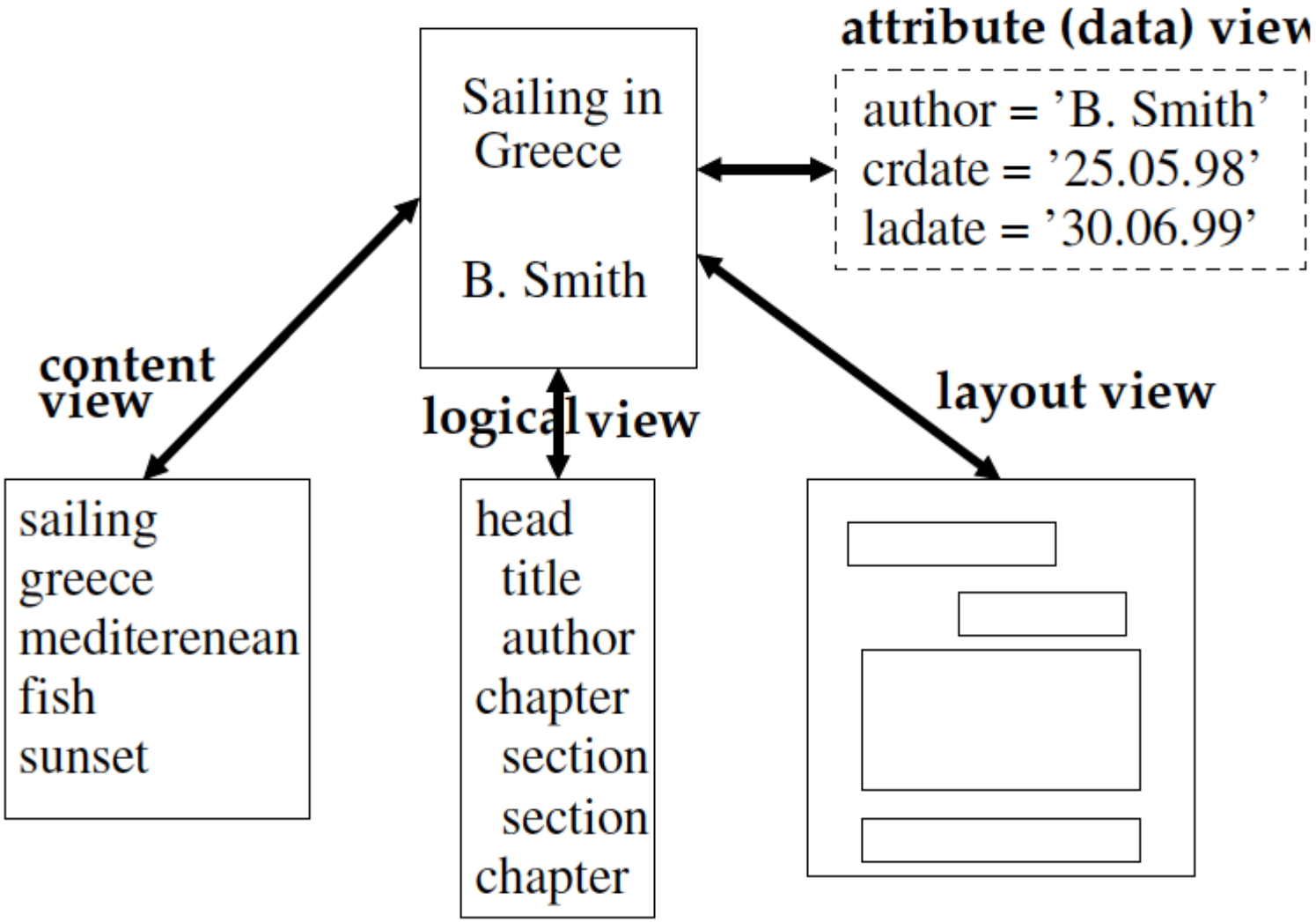
- Unit of retrieval and Free text:
  - text, strings of characters from an alphabet
  - natural language:
  - newspaper articles, journal paper, dictionary denition, e-mailmessages
  - size of documents:
    - arbitrary, newspaper article vs journal article vs e-mail
- Sub-document can also be a unit of retrieval (XML element, answer to a question)

History

- Manual IR in libraries: manual indexing; manual categorisation
- 70ies and 80ies: Automatic IR in libraries
- 90ies: IR on the web and in digital libraries
- Success factors: Response time, coverage, interactivity, low (no!) costs, precision-oriented

$$\text{precision} \approx \text{correctness}, \quad \text{recall} \approx \text{completeness}$$

Document Indexing



Document Indexing – Vector Space Model

- Language used to describe documents and queries
  - Index terms
- Linear Algebraic model for representing text documents

Vector Space Model

- Documents and queries are represented as vectors.
- Each dimension corresponds to a separate term. If a term (or keyword) occurs in the document, its value in the vector is non-zero.

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$$

- TF-IDF (Term Frequency Inverse Document Frequency)
  - computes the values of the terms (or keywords) weights taking into account the frequency of the term in a document and in the whole corpus.

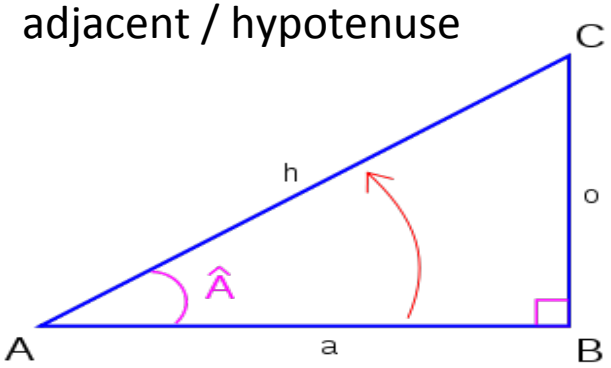
Relevance Ranking - Similarity

- Useful to compute de similarity between a query and a document or among documents.
- Compares the deviation of angles between each document vector and the original query vector, where the query is represented as the same kind of vector as the documents.
- It is easier to calculate the cosine of the angle between the vectors, instead of the angle itself.

$$\cos\theta = \frac{\mathbf{d}_2 \cdot \mathbf{q}}{\|\mathbf{d}_2\| \, \|\mathbf{q}\|}$$

Cosinus

adjacent / hypotenuse



$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

TF-IDF Hypothesis

Frequency of words is a measurement of a word signicance.

... a measurement of the power of a word to discriminate documents by their content ...

Stop Words

- is, a, the, or, and, ...
- not?
- other?
- Stop-word list often dened manually.
- Reduction: between 30 and 50 per cent.

Index Term Weighting

- Exhaustivity
  - number of different topics indexed
  - importance of term in a document
  - high exhaustivity: high recall and low precision
- Specifity
  - ability of the indexing technique to describe topics precisely
  - number of documents to which a term is assigned in a collection
  - related to the distribution of index terms in collection
  - high specifcity: high precision and low recall

Index Term Weighting

- Index term weighting
  - index term frequency: occurrence frequency of a term in document
  - document frequency: number of documents in which a term occurs

TF-IDF

$weight(t,d) = tf(t,d) \times idf(t)$	
N	number of documents in collection
$n(t)$	number of documents in which term t occurs
$idf(t)$	inverse document frequency of term t
$occ(t,d)$	occurrence of term t in document d
t_{max}	term in document d with highest occurrence
$tf(t,d)$	term frequency of t in document d

$$tf(t,d) := \frac{occ(t,d)}{occ(t_{max},d)} \qquad idf(t) := \log \frac{N}{n(t)}$$

Example

Nuclear 7	Computer 9
Poverty 5	Unemployment 1
Luddites 3	Machines 19
People 25	And 49

Weight(machine) =
19/25 x log(100/50)
= 0.76 x 0.3013 = 0.228988

Weight(luddite) =
3/25 x log(100/2)
= 0.12 x 1.69897 = 0.2038764

Weight(poverty) = 5/25 x log(100/2) = 0.2 x 1.69897 = 0.339794

Inverted Index

- In computer science, an inverted index is an index data structure storing a mapping from content, such as words or numbers, to its locations in a database file. The purpose of an inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database (Wikipedia)

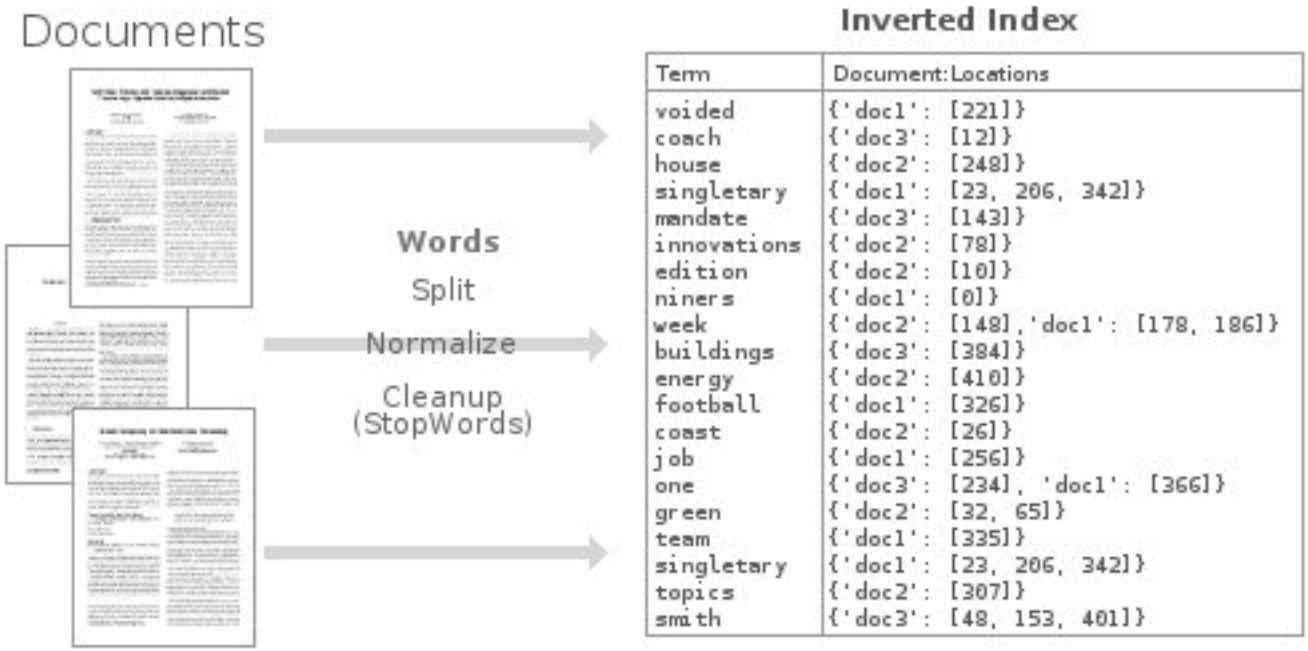
Documents:

D0 = "it is what it is"
D1 = "what is it"
D2 = "it is a banana"

Inverted Index:

"a": {2}
"banana": {2}
"is": {0, 1, 2}
"it": {0, 1, 2}
"what": {0, 1}

Inverted Index



Queries

- Simple queries: key words
- Best Match (query, documents)
  - Compare the terms in a document and query
  - Compute "similarity » (cosinus) between each document in the collection and the query based on the terms they have in common
  - Sorting the document in order of decreasing similarity with the query
  - The outputs are a ranked list and displayed to the user - the top ones are more relevant as judged by the system

Query Processing – Top-k processing

Top-k query processing

=

Finding k documents that have the highest overall grade wrt to the score:

$COS(Q,D_i)$, considering the whole corpus of documents.

In other words inspect the inverted index using for instant a Fagin's Algorithm

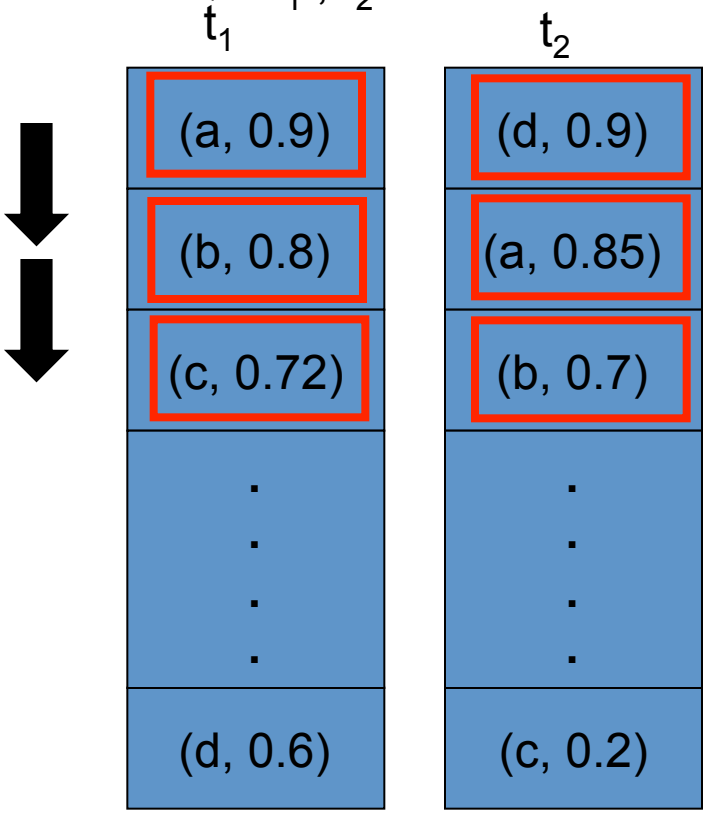
Top-k processing

- Input: inverted index, query vector, similarity mesure (cosinus)
- Output: top-k documents

Example – Fagin's Algorithm

STEP 1

- k = 3 and Q = t_1 , t_2
- Read *tf-idf* weights (w_i) from every term in the sorted inverted lists (in parallel)
- Stop when *k* documents have been seen **in common from all lists**
- Ex: Return the the k=3 most relevant documents (a, b, c, etc) for the query and Q = t_1 , t_2



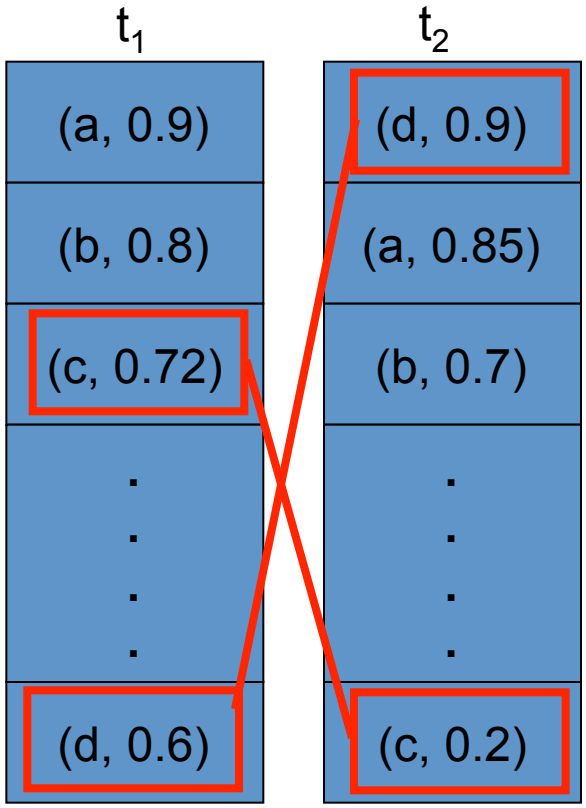
Docs	w_1	w_2	Cos(Q,d_i)
a	0.9	0.85	
d		0.9	
b	0.8	0.7	
c	0.72		

Inverted Indexs are ordered in decreasing order (monotonicity)

Example – Fagin's Algortihm

STEP 2

- Random access to find missing term weights.



Docs	w_1	w_2	$Cos(Q,d_i)$
a	0.9	0.85	
d	0.6	0.9	
b	0.8	0.7	
c	0.72	0.2	

Example – Fagin's Algortihm

STEP 3

- Compute the Cosinus score of the seen documents.
- Return the k highest scored documents.

t_1	t_2
(a, 0.9)	(d, 0.9)
(b, 0.8)	(a, 0.85)
(c, 0.72)	(b, 0.7)
. . . .	. . . .
(d, 0.6)	(c, 0.2)

Docs	w_1	w_2	$Cos(Q,d_i)$
a	0.9	0.85	0.85
d	0.6	0.9	0.6
b	0.8	0.7	0.7
c	0.72	0.2	0.2

Advantages

- Simple model based on linear algebra
- Term weights not binary
- Allows computing a continuous degree of similarity between queries and documents
- Allows ranking documents according to their possible relevance
- Allows partial matching