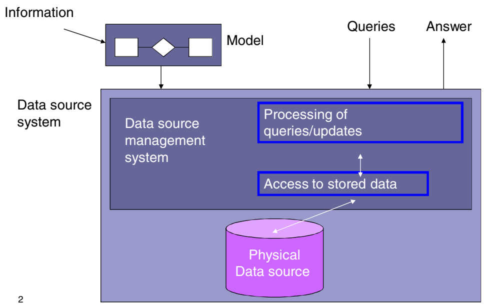
Information Retrieval

Patrick Lambrix
Department of Computer and Information Science
Linköpings universitet

Data sources

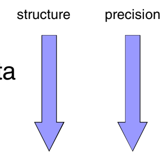


Storing and accessing textual information

- How is the information stored?
  - high level
- How is the information retrieved?

Storing textual information

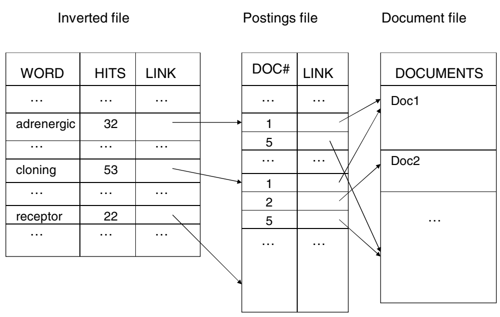
- Text (IR)
- Semi-structured data
- Data models (DB)
- Rules + Facts (KB)



Storing textual information - Text - Information Retrieval

- search using words
- conceptual models:
  boolean, vector, probabilistic, …
- file model:
  flat file, inverted file, ...

IR - File model: inverted files



IR – File model: inverted files

- Controlled vocabulary
- Stop list
- Stemming

7

IR - formal characterization

Information retrieval model: (D,Q,F,R)

- D is a set of document representations
- Q is a set of queries
- F is a framework for modeling document representations, queries and their relationships
- R associates a real number to document-query-pairs (ranking)

8

IR - conceptual models

Classic information retrieval

- Boolean model
- Vector model
- Probabilistic model

9

Boolean model

Document representation

	adrenergic	cloning	receptor		
Doc1	yes	yes	no	-->	(1 1 0)
Doc2	no	yes	no	-->	(0 1 0)

10

Boolean model

Queries : boolean (and, or, not)

Q1: cloning and (adrenergic or receptor)

Queries are translated to disjunctive normal form (DNF)

DNF: disjunction of conjunctions of terms with or without 'not'

Rules: not not A --> A
not(A and B) --> not A or not B
not(A or B) --> not A and not B
(A or B) and C --> (A and C) or (B and C)
A and (B or C) --> (A and B) or (A and C)
(A and B) or C --> (A or C) and (B or C)
A or (B and C) --> (A or B) and (A or C)

11

Boolean model

Q1: cloning and (adrenergic or receptor)
--> (cloning and adrenergic) or (cloning and receptor)

DNF is completed
+ translated to same representation as documents

(cloning and adrenergic) or (cloning and receptor)
--> (cloning and adrenergic and receptor)
or (cloning and adrenergic and not receptor)
or (cloning and receptor and adrenergic)
or (cloning and receptor and not adrenergic)
--> (1 1 1) or (1 1 0) or (1 1 1) or (0 1 1)
--> (1 1 1) or (1 1 0) or (0 1 1)

12

Boolean model

	adrenergic	cloning	receptor		
Doc1	yes	yes	no	-->	(1 1 0)
Doc2	no	yes	no	-->	(0 1 0)

Q1: cloning and (adrenergic or receptor)

--> (1 1 0) or (1 1 1) or (0 1 1)   Result: Doc1

Q2: cloning and not adrenergic

--> (0 1 0) or (0 1 1)   Result: Doc2

Boolean model

Advantages

- based on intuitive and simple formal model (set theory and boolean algebra)

Disadvantages

- binary decisions
  - words are relevant or not
  - document is relevant or not, no notion of partial match

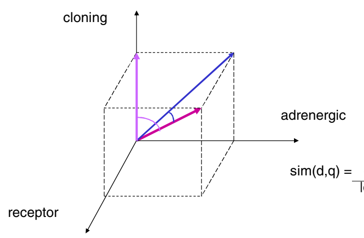
Boolean model

	adrenergic	cloning	receptor		
Doc1	yes	yes	no	-->	(1 1 0)
Doc2	no	yes	no	-->	(0 1 0)

Q3: adrenergic and receptor

--> (1 0 1) or (1 1 1)   Result: empty

Vector model (simplified)



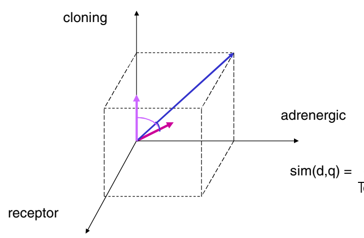
Doc1 (1,1,0)
Doc2 (0,1,0)
Q (1,1,1)

$sim(d,q) = \frac{d \cdot q}{|d| \times |q|}$

Vector model

- Introduce weights in document vectors (e.g. Doc3 (0, 0.5, 0))
- Weights represent importance of the term for describing the document contents
- Weights are positive real numbers
- Term does not occur -> weight = 0

Vector model



Doc1 (1,1,0)
Doc3 (0,0.5,0)
Q4 (0.5,0.5,0.5)

$sim(d,q) = \frac{d \cdot q}{|d| \times |q|}$

Vector model

- How to define weights? tf-idf

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

$w_{i,j}$ = weight for term k_i in document d_j

$= f_{i,j} \times idf_i$

19

Vector model

- How to define weights? tf-idf

term frequency $freq_{i,j}$: how often does term k_i occur in document d_j?

normalized term frequency:

$f_{i,j} = freq_{i,j} / \max_l freq_{l,j}$

20

Vector model

- How to define weights? tf-idf

document frequency : in how many documents does term k_i occur?

N = total number of documents

n_i = number of documents in which k_i occurs

inverse document frequency idf_i: $\log (N / n_i)$

21

Vector model

- How to define weights for query?

recommendation:

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

$w_{i,q}$ = weight for term k_i in q

$= (0.5 + 0.5\, f_{i,q}) \times idf_i$

22

Vector model

- Advantages
- term weighting improves retrieval performance
- partial matching
- ranking according to similarity

Disadvantage

- assumption of mutually independent terms?

23

Probabilistic model

weights are binary ($w_{i,j} = 0$ or $w_{i,j} = 1$)

R: the set of relevant documents for query q

Rc: the set of non-relevant documents for q

P(R|dj): probability that dj is relevant to q

P(Rc|dj): probability that dj is not relevant to q

$sim(d_j,q) = P(R|d_j) / P(Rc|d_j)$

24

Probabilistic model

sim(dj,q) = P(R|dj) **/** P(Rc|dj)

(Bayes' rule, independence of index terms, take logarithms, P(ki|R) + P(not ki|R) = 1)

--> SIM(dj,q) ==

SUM $_{i=1}^{t}$ wi,q x wi,j x

(log(P(ki|R) **/** (1- P(ki|R))) +

log((1- P(ki|Rc)) / P(ki|Rc)))

25

Probabilistic model

- How to compute P(ki|R) and P(ki|Rc)?

- initially: P(ki|R) = 0.5 and P(ki|Rc) = ni**/**N

- Repeat: retrieve documents and rank them

V: subset of documents (e.g. r best ranked)

Vi: subset of V, elements contain ki

P(ki|R) = |Vi| **/** |V|

and P(ki|Rc) = (ni-|Vi|**)** **/(**N-|V|)

26

Probabilistic model

- Advantages:

- ranking of documents with respect to probability of being relevant

- Disadvantages:

- initial guess about relevance

- all weights are binary

- independence assumption?

27

IR - measures

Precision =

$$\frac{\text{number of found relevant documents}}{\text{total number of found documents}}$$

Recall =

$$\frac{\text{number of found relevant documents}}{\text{total number of relevant documents}}$$

28

Literature

Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval,* Addison-Wesley, 1999.

29