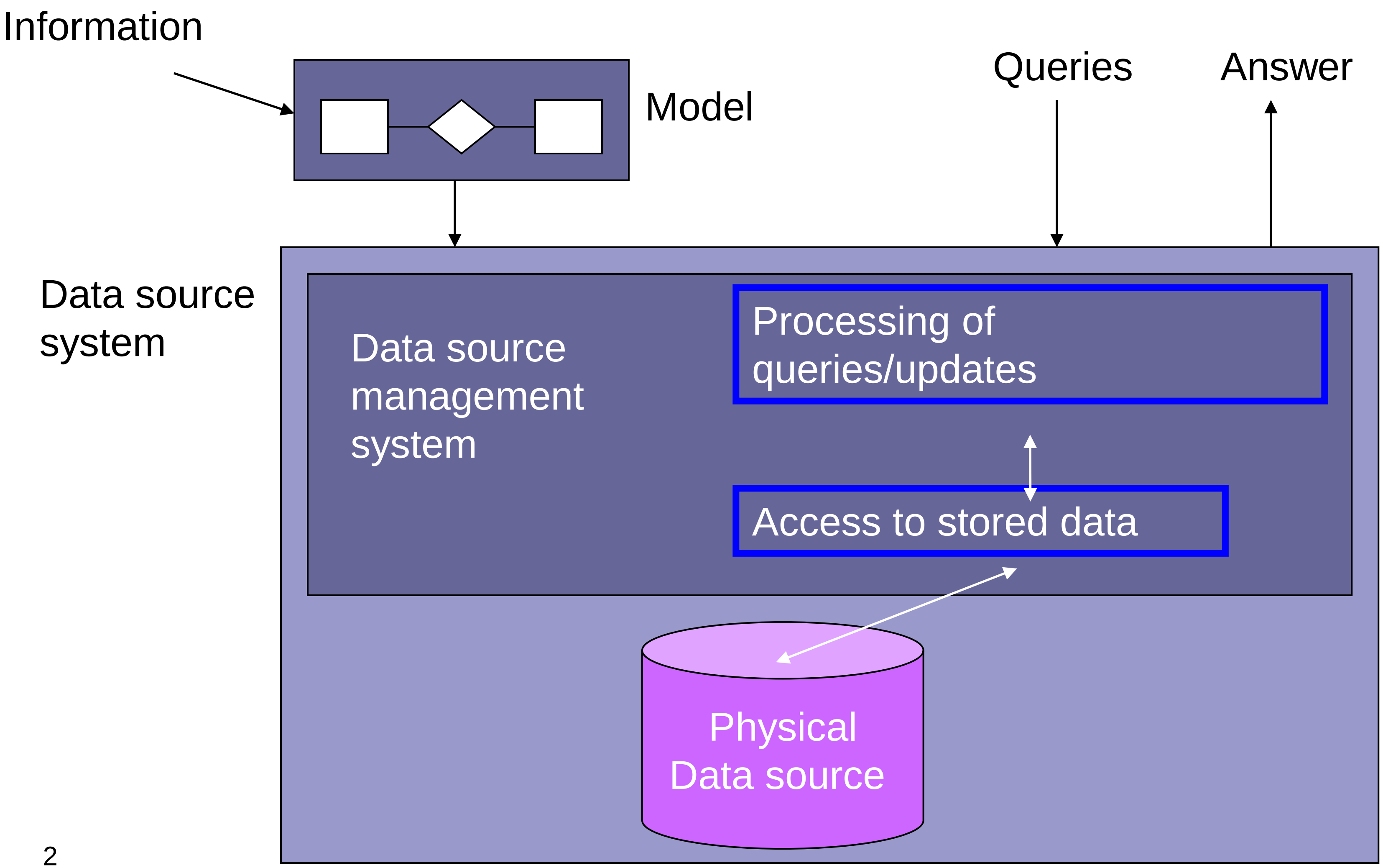
Information Retrieval

Patrick Lambrix
Department of Computer and Information Science
Linköpings universitet

Data sources

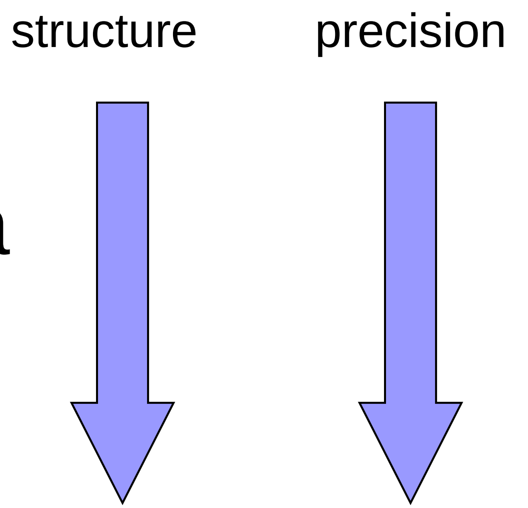


Storing and accessing textual information

- 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

Inverted file

WORD	HITS	LINK
…	…	…
adrenergic	32	
…	…	…
cloning	53	
…	…	…
receptor	22	
…	…	…

Postings file

DOC#	LINK
…	…
1	
5	
…	…
1	
2	
5	
…	…

Document file

DOCUMENTS
Doc1
Doc2
…

IR – File model: inverted files

- Controlled vocabulary
- Stop list
- Stemming

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)

IR - conceptual models

Classic information retrieval

- Boolean model
- Vector model
- Probabilistic model

IR - conceptual models - Summary

	Boolean	Vector	Probabilistic
D			
Q			
F			
R			

IR - conceptual models - Summary

Boolean

D

Q

F

R

Boolean model

Document representation

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

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’

Boolean model

DNF or not DNF?

1. (A and B) or (C and D)
2. (A and B and C) or (A and D) or (E and F)
3. (A or B) and (C or D)
4. (A and not B) or (C and D)
5. (A and not B) or not(A and B)
6. (not not A and B)
7. A and B
8. A or B
9. A
10. not A
11. not not A

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)

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)

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

IR - conceptual models - Summary

Vector simplified / Vector

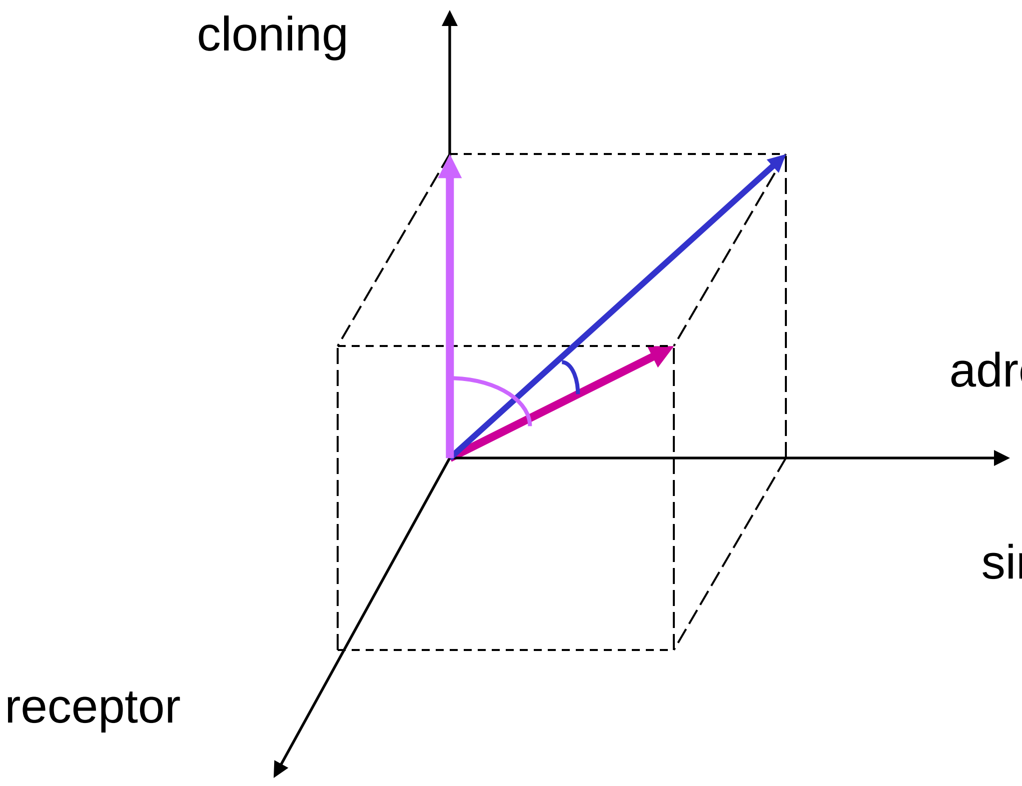
D

Q

F

R

Vector model (simplified)



Doc1 (1,1,0)

Doc2 (0,1,0)

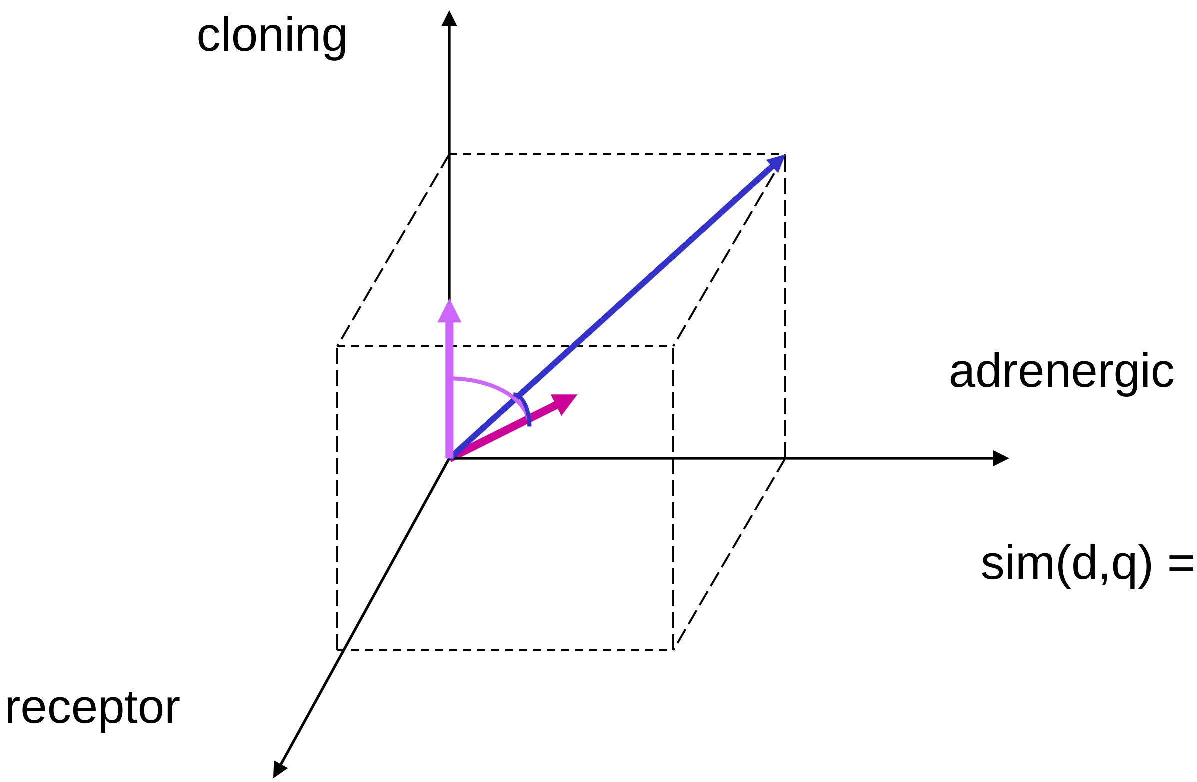
Q (1,1,1)

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

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)

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

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

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

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:

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

Vector model

Example:

	Doc1	Doc2
K1: adrenergic	5	0
K2: cloning	0	10
K3: receptor	20	2

freqi,j

Vector model

Example:

	Doc1	Doc2
K1: adrenergic	5/20	0/10
K2: cloning	0/20	10/10
K3: receptor	20/20	2/10

$freq_{i,j} \rightarrow f_{i,j}$

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: $\log_2 (N / n_i)$

Vector model

Example:

	Doc1	Doc2	idfi
K1: adrenergic	5/20	0/10	1
K2: cloning	0/20	10/10	1
K3: receptor	20/20	2/10	0

$\log_2 (2/1) = \log_2 2 = 1; \log_2 (2/2) = \log_2 1 = 0$

Vector model

Example:

	Doc1	Doc2	idfi
K1: adrenergic	5/20	0/10	1
K2: cloning	0/20	10/10	1
K3: receptor	20/20	2/10	0
	(0.25,0,0)	(0,1,0)	

Vector model

- How to define weights for query?

recommendation:

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

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

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

Vector model

- Advantages

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

Disadvantage

- assumption of mutually independent terms?

IR - conceptual models - Summary

Probabilistic

D

Q

F

R

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

$$sim(d_j,q) = P(R|d_j) \mathbin{\boldsymbol{/}} P(Rc|d_j)$$

Probabilistic model

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

(Bayes' rule, independence of index terms, take logarithms, $P(k_i|R) + P(\text{not } k_i|R) = 1$)

--> SIM(dj,q) ==

$$SUM_{i=1}^{t} \; Wi,q \; X \; Wi,j \; X$$

$$(\log(P(k_i|R) \,/\, (1- P(k_i|R))) +$$

$$\log((1- P(k_i|Rc)) \,/\, P(k_i|Rc)))$$

Probabilistic model

- How to compute $P(k_i|R)$ and $P(k_i|R^c)$?

  - initially: $P(k_i|R) = 0.5$ and $P(k_i|R^c) = n_i/N$

    with N = number of documents and

    n_i = number of documents containing k_i

  - Repeat: retrieve documents and rank them

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

   V_i: subset of V, elements contain k_i

   $P(k_i|R) = |V_i| / |V|$

   and $P(k_i|R^c) = (n_i-|V_i|) / (N-|V|)$

Probabilistic model

- Advantages:

- ranking of documents with respect to probability of being relevant

- Disadvantages:

- initial guess about relevance

- all weights are binary

- independence assumption?

IR - measures

Precision =

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

Recall =

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

IR – measures (visual)

Literature

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