Lecture 3 Modeling

Ranking

- central problem of IR
  - Predict which documents are relevant and which are not
- Ranking
  - Establish an ordering of the documents retrieved
- IR models
  - Different model provides distinct sets of premises to deal with document relevance

Information Retrieval Models

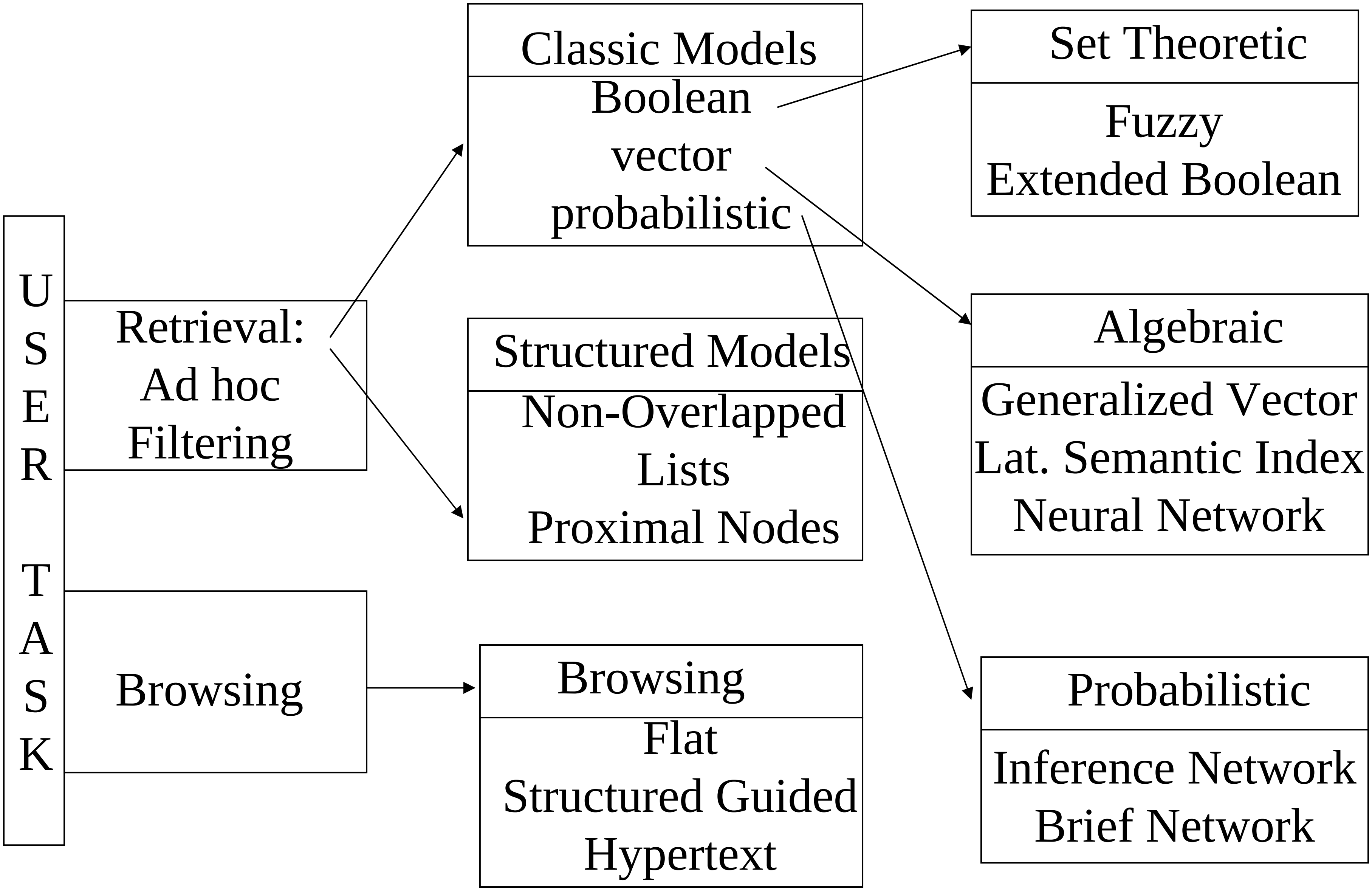
- Classic Models
  - Boolean model
    - set theoretic
    - documents and queries are represented as sets of index terms
    - compare Boolean query statements with the term sets used to identify document content.
  - Vector model
    - algebraic model
    - documents and queries are represented as vectors in a t-dimensional space
    - compute global similarities between queries and documents.
  - Probabilistic model
    - probabilistic
    - documents and queries are represented on the basis of probabilistic theory
    - compute the relevance probabilities for the documents of a collection.

Information Retrieval Models

(Continued)

- Structured Models
  - reference to the structure present in written text
  - non-overlapping list model
  - proximal nodes model
- Browsing
  - flat
  - structured guided
  - hypertext

Taxonomy of Information Retrieval Models



USER TASK

Retrieval:
Ad hoc
Filtering

Browsing

Classic Models

Boolean
vector
probabilistic

Structured Models

Non-Overlapped Lists
Proximal Nodes

Browsing

Flat
Structured Guided
Hypertext

Set Theoretic

Fuzzy
Extended Boolean

Algebraic

Generalized Vector
Lat. Semantic Index
Neural Network

Probabilistic

Inference Network
Brief Network

Issues of a retrieval system

- Models
  - Boolean
  - vector
  - probabilistic
- Logical views of documents
  - full text
  - set of index terms
- User task
  - retrieval
  - browsing

Combinations of these issues

LOGICAL VIEW OF DOCUMENTS

USER TASK

	Index Terms	**Full Text**	**Full Text+ Structure**
Retrieval	Classic Set Theoretic Algebraic Probabilistic	Classic Set Theoretic Algebraic Probabilistic	Structured
Browsing	Flat	Flat Hypertext	Structure Guided Hypertext

Retrieval: Ad hoc and Filtering

- Ad hoc retrieval
  - Documents remain relatively static while new queries are submitted
- Filtering
  - Queries remain relatively static while new documents come into the system
    - e.g., news wiring services in the stock market
  - User profile describes the user’s preferences
    - Filtering task indicates to the user which document might be interested to him
    - Which ones are really relevant is fully reserved to the user
  - Routing: a variation of filtering
    - Ranking filtered documents and show this ranking to users

User profile

- Simplistic approach
  - The profile is described through a set of keywords
  - The user provides the necessary keywords
- Elaborate approach
  - Collect information from the user
  - initial profile + relevance feedback (relevant information and nonrelevant information)

Formal Definition of IR Models

- /D, Q, F, R(q_i, d_j)/
  - D: a set composed of logical views (or representations) for the documents in collection
  - Q: a set composed of logical views (or representations) for the user information needs

query

  - F: a framework for modeling documents representations, queries, and their relationships
  - R(q_i, d_j): a ranking function which associations a real number with $q_i \in Q$ and $d_j \in D$

Formal Definition of IR Models

(*continued*)

- classic Boolean model
  - – set of documents
  - – standard operations on sets
- classic vector model
  - – t-dimensional vector space
  - – standard linear algebra operations on vector
- classic probabilistic model
  - – sets
  - – standard probabilistic operations, and Bayes’ theorem

Basic Concepts of Classic IR

- index terms (usually nouns): index and summarize
- weight of index terms
- Definition
  - $K=\{k_1, \ldots, k_t\}$: a set of all index terms
  - $w_{i,j}$: a weight of an index term k_i of a document d_j
  - $\vec{d}_j=(w_{1,j}, w_{2,j}, \ldots, w_{t,j})$: an *index term vector* for the document d_j
  - $g_i(\vec{d}_j)= w_{i,j}$

$w_{i,j}$ associated with (k_i,d_j) tells us nothing about $w_{i+1,j}$ associated with (k_{i+1},d_j)

- assumption
  - index term weights are *mutually independent*

The terms *computer* and *network* in the area of computer networks

Boolean Model

Boolean Model

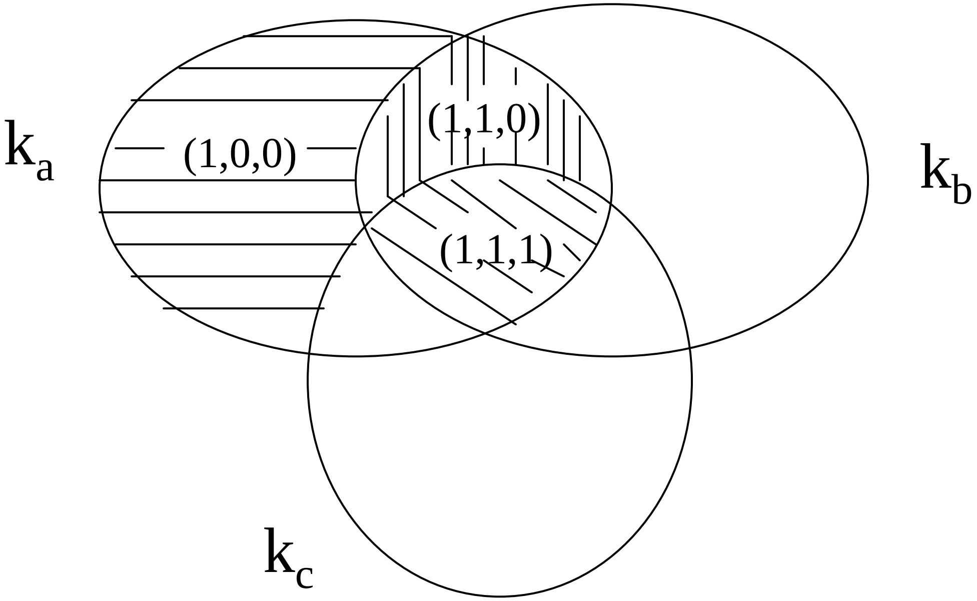
- The index term weight variables are all binary, i.e., $w_{i,j} \in \{0,1\}$
- A query q is a Boolean expression (and, or, not)
- $\vec{q}_{dnf}$: the *disjunctive normal form* for q
- $\vec{q}_{cc}$: conjunctive components of $\vec{q}_{dnf}$
- sim(d_j,q): similarity of d_j to q
  - 1: if $\exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf} \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc}))$
  - 0: otherwise

dj is relevant to q

Boolean Model (*Continued*)

$(k_a \wedge k_b) \vee (k_a \wedge \neg k_c)$
$= (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c)$
$\vee (k_a \wedge k_b \wedge \neg k_c) \vee (k_a \wedge \neg k_b \wedge \neg k_c)$
$= (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c) \vee$
$(k_a \wedge \neg k_b \wedge \neg k_c)$

- Example
  - $q = k_a \wedge (k_b \vee \neg k_c)$
  - $\overrightarrow{q_{dnf}} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$



Boolean Model (*Continued*)

- advantage: simple
- disadvantage
  - binary decision (relevant or non-relevant) without grading scale
  - exact match (no partial match)
    - e.g., $\vec{d}_j=(0,1,0)$ is non-relevant to $q=k_a \wedge (k_b \vee \neg k_c)$
  - retrieve too few or too many documents

Vector Model

Basic Vector Space Model

- *Term vector* representation of

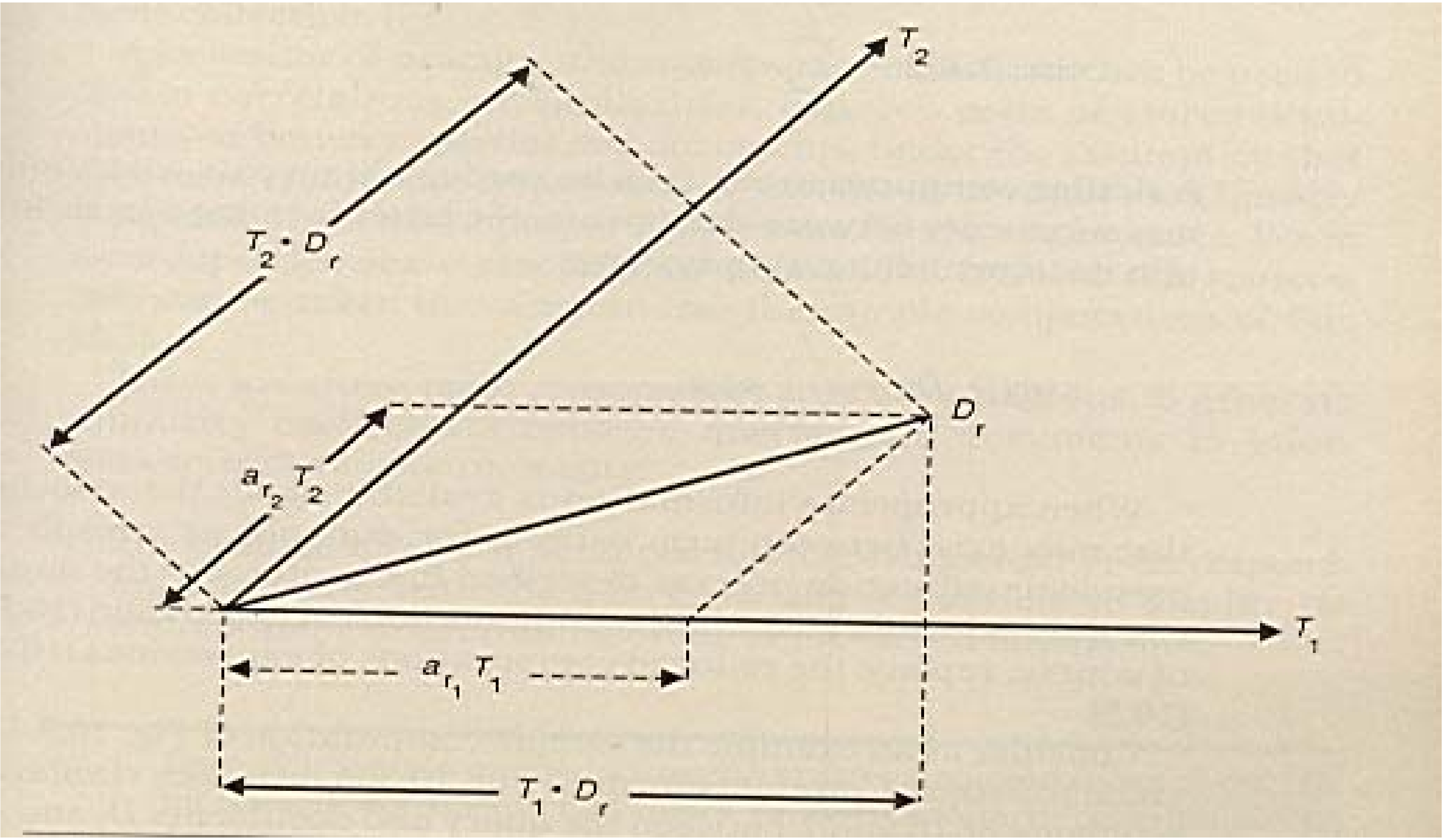
  documents $D_i=(a_{i1}, a_{i2}, \ldots, a_{it})$

  queries $Q_j=(q_{j1}, q_{j2}, \ldots, q_{jt})$
- t distinct terms are used to characterize content.
- Each term is identified with a term vector T.
- t vectors are linearly independent.
- Any vector is represented as a linear combination of the t term vectors.
- The rth document D_r can be represented as a document vector, written as

$$D_r = \sum_{i=1}^{t} a_{ri} T_i$$

Document representation in vector space

a document vector in a two-dimensional vector space



Similarity Measure

- measure by product of two vectors

  $$x \bullet y = |x|\,|y| \cos\alpha$$

- document-query similarity

document vector:

$$D_r = \sum_{i=1}^{t} a_{ri} T_i$$

term vector:

$$Q_s = \sum_{j=1}^{t} q_{sj} T_j$$

$$D_r \bullet Q_s = \sum_{i,j=1}^{t} a_{ri} q_{sj} T_i \bullet T_j$$

- how to determine the vector components and term correlations?

Similarity Measure (*Continued*)

- vector components

$$A = \begin{array}{c|cccc|}
 & T_1 & T_2 & T_3 & T_t \\
D_1 & a_{11} & a_{12} & \cdots & a_{1t} \\
D_2 & a_{21} & a_{22} & \cdots & a_{2t} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
D_n & a_{n1} & a_{n2} & \cdots & a_{nt}
\end{array}$$

Similarity Measure (*Continued*)

- term correlations $T_i \bullet T_j$ are not available
  assumption: term vectors are orthogonal

$$T_i \bullet T_j = 0 \ (i \neq j) \quad T_i \bullet T_j = 1 \ (i = j)$$

- Assume that terms are uncorrelated.

$$sim(D_r, Q_s) = \sum_{i,j=1}^{t} a_{ri} q_{sj}$$

- Similarity measurement between documents

$$sim(D_r, D_s) = \sum_{i,j=1}^{t} a_{ri} a_{sj}$$

Sample query-document similarity computation

- $D_1=2T_1+3T_2+5T_3$ $D_2=3T_1+7T_2+1T_3$
  $Q=0T_1+0T_2+2T_3$
- similarity computations for uncorrelated terms
  $sim(D_1,Q)=2•0+3•0+5•2=10$
  $sim(D_2,Q)=3•0+7•0+1•2=2$
- D_1 is preferred

Sample query-document similarity computation (*Continued*)

-

	T_1	T_2	T_3
T_1	1	0.5	0
T_2	0.5	1	-0.2
T_3	0	-0.2	1

- similarity computations for correlated terms

  $sim(D_1,Q)=(2T_1+3T_2+5T_3) \bullet (0T_1+0T_2+2T_3\ )$

  $=4T_1 \bullet T_3+6T_2 \bullet T_3+10T_3 \bullet T_3$

  $=-6*0.2+10*1=8.8$

  $sim(D_2,Q)=(3T_1+7T_2+1T_3) \bullet (0T_1+0T_2+2T_3\ )$

  $=6T_1 \bullet T_3+14T_2 \bullet T_3+2T_3 \bullet T_3$

  $=-14*0.2+2*1=-0.8$

- D_1 is preferred

Vector Model

- $w_{i,j}$: a positive, *non-binary weight* for (k_i,d_j)
- $w_{i,q}$: a positive, *non-binary weight* for (k_i,q)
- $\vec{q}=(w_{1,q}, w_{2,q}, \ldots, w_{t,q})$: a query vector, where t is the total number of index terms in the system
- $\vec{d}_j= (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$: a document vector

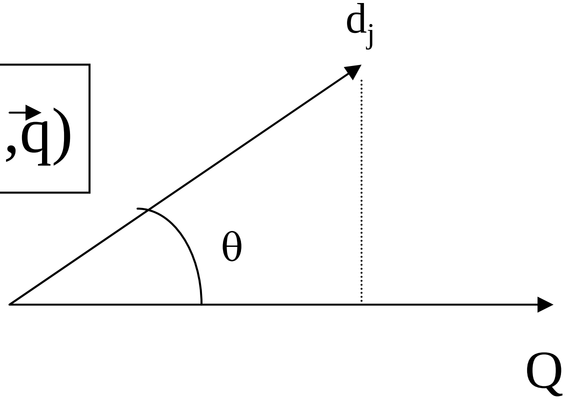
Similarity of document d_j w.r.t. query q

- The correlation between vectors $\vec{d_j}$ and $\vec{q}$

$$sim(d_j, q) = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|}$$

$$= \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$$

$$\boxed{\cos(\vec{d_j}, \vec{q})}$$



- $|\vec{q}|$ does not affect the ranking
- $|\vec{d_j}|$ provides a normalization

document ranking

- Similarity (i.e., sim(q, d_j)) varies from 0 to 1.
- Retrieve the documents with a degree of similarity above a predefined threshold (allow partial matching)

term weighting techniques

- IR problem: one of clustering
  - user query: a specification of a set A of objects
  - clustering problem: determine which documents are in the set A (*relevant*), which ones are not (*non-relevant*)
  - intra-cluster similarity
    - the features better describe the objects in the set A
    - tf factor in vector model
      the raw frequency of a term k_i inside a document d_j
  - inter-cluster dissimilarity
    - the features better distinguish the the objects in the set A from the remaining objects in the collection C
    - idf factor (inverse document frequency) in vector model
      the inverse of the frequency of a term k_i among the documents in the collection

Definition of *tf*

- N: total number of documents in the system
- n_i: the number of documents in which the index term k_i appears
- $freq_{i,j}$: the raw frequency of term k_i in the document d_j
- $f_{i,j}$: the *normalized frequency* of term k_i in document d_j

(0~1)

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

Term t_l has maximum frequency in the document d_j

Definition of *idf* and *tf-idf* scheme

- idf_i: inverse document frequency for k_i

$$idf_i = \log \frac{N}{n_i}$$

- $w_{i,j}$: term-weighting by *tf-idf* scheme

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

- *query term* weight (Salton and Buckley)

(a very short document) $$w_{i,q} = (0.5 + \frac{0.5\, freq_{i,q}}{\max_l\ freq_{i,q}}) \times \log \frac{N}{n_i}$$

$\mathrm{freq}_{i,q}$: the raw frequency of the term k_i in q

Analysis of vector model

- advantages
  - its *term-weighting* scheme improves *retrieval performance*
  - its *partial matching* strategy allows retrieval of documents that *approximate* the query conditions
  - its *cosine ranking* formula sorts the documents according to their *degree of similarity* to the query
- disadvantages
  - indexed terms are assumed to be *mutually independently*

Probabilistic Model

Probabilistic Model

- Given a query, there is an *ideal answer set*
  - a set of documents which contains exactly the relevant documents and no other
- query process
  - a process of specifying *the properties* of an ideal answer set
- problem: what are the properties?

Probabilistic Model (*Continued*)

- Generate a preliminary probabilistic description of the ideal answer set
- Initiate an interaction with the user
  - User looks at the retrieved documents and decide which ones are relevant and which ones are not
  - System uses this information to refine the description of the ideal answer set
  - Repeat the process many times.

Probabilistic Principle

- Given a *user query* q and a *document* d_j in the collection, the probabilistic model estimates the probability that user will find d_j relevant
- assumptions
  - The probability of relevance depends on query and document representations only
  - There is a subset of all documents which the user prefers as the answer set for the query q
- Given a query, the probabilistic model assigns to each document dj a measure of its similarity to the query

$$\frac{P(d_j \; relevant - to \; q)}{P(d_j \; nonrelevant - to \; q)}$$

Probabilistic Principle

- $w_{i,j} \in \{0,1\}$, $w_{i,q} \in \{0,1\}$: the index term weight variables are all binary non-relevant
- q: a query which is a subset of index terms
- R: the set of documents known to be *relevant*
- $\overline{R}$ (complement of R): the set of *non-relevant* documents
- $P(R|\vec{d_j})$: the probability that the document d_j is *relevant* to the query q
- $P(\overline{R}|\vec{d_j})$: the probability that d_j is *non-relevant* to q

similarity

- sim(d_j,q): the similarity of the document d_j to the query q

$$sim(d_j,q) = \frac{P(R \mid \vec{d}_j)}{P(\overline{R} \mid \vec{d}_j)}$$

(by definition)

$$sim(d_j,q) = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \overline{R}) \times P(\overline{R})}$$

(Bayes' rule) $P(X \mid Y) = \frac{P(X)P(Y \mid X)}{P(Y)}$

$$sim(d_j,q) \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \overline{R})}$$

(P(R) and P($\overline{R}$) are the same for all documents)

$P(\vec{d}_j \mid R)$: the probability of randomly selecting the document d_j from the set of R of relevant documents

$P(R)$: the probability that a document randomly selected from the entire collection is relevant

$$sim(d_j, q) \approx \frac{P(\overrightarrow{d_j} \mid R)}{P(\overrightarrow{d_j} \mid \overline{R})}$$

$$= \log \frac{\prod_{i=1}^{t} (P(k_i \mid R))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid R))^{1-g_i(\overrightarrow{d_j}) g_i(\vec{q})}}{\prod_{i=1}^{t} (P(k_i \mid \overline{R}))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid \overline{R}))^{1-g_i(\overrightarrow{d_j}) g_i(\vec{q})}}$$

independence assumption of index terms

$$= \sum_{i=1}^{t} \log \frac{(P(k_i \mid R))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid R))^{1-g_i(\overrightarrow{d_j}) g_i(\vec{q})}}{(P(k_i \mid \overline{R}))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid \overline{R}))^{1-g_i(\overrightarrow{d_j}) g_i(\vec{q})}}$$

$$= \sum_{i=1}^{t} \log \frac{(P(k_i \mid R) \times P(\overline{k}_i \mid \overline{R}))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid R))}{(P(k_i \mid \overline{R}) \times P(\overline{k}_i \mid R))^{g_i(\overrightarrow{d_j}) g_i(\vec{q})} \times (P(\overline{k}_i \mid \overline{R}))}$$

$$= \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times \log \frac{P(k_i \mid R) \times P(\overline{k}_i \mid \overline{R})}{P(k_i \mid \overline{R}) \times P(\overline{k}_i \mid R)} + \sum_{i=1}^{t} \log \frac{P(\overline{k}_i \mid R)}{P(\overline{k}_i \mid \overline{R})}$$

$$= \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times \log \frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))} + \sum_{i=1}^{t} \log \frac{P(\overline{k}_i \mid R)}{P(\overline{k}_i \mid \overline{R})}$$

$$sim(d_j, q) \approx \frac{P(\overrightarrow{d_j} \mid R)}{P(\overrightarrow{d_j} \mid \overline{R})}$$

$$= \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times \log \frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))} + \sum_{i=1}^{t} \log \frac{P(\overline{k}_i \mid R)}{P(\overline{k}_i \mid \overline{R})}$$

$$= \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times (\log \frac{P(k_i \mid R)}{(1 - P(k_i \mid R))}) + \log \frac{(1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R})}) + \sum_{i=1}^{t} \log \frac{P(\overline{k}_i \mid R)}{P(\overline{k}_i \mid \overline{R})}$$

$$\approx \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times (\log \frac{P(k_i \mid R)}{(1 - P(k_i \mid R))}) + \log \frac{(1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R})})$$

Problem: where is the set R?

Initial guess

- $P(k_i|R)$ is constant for all index terms k_i.

$$p(k_i \mid R) = 0.5$$

- The distribution of index terms among the non-relevant documents can be approximated by the distribution of index terms among all the documents in the collection.

$$P(k_i \mid \overline{R}) = \frac{n_i}{N}$$

(假設$N >> |R|, N \approx |\overline{R}|$)

Initial ranking

- V: a subset of the documents initially retrieved and ranked by the probabilistic model (*top r documents*)
- V_i: subset of V composed of documents which contain the index term k_i
- Approximate $P(k_i|R)$ by the distribution of the index term k_i among the documents retrieved so far.

$$P(k_i \mid R) = \frac{V_i}{V}$$

- Approximate $P(k_i|\overline{R})$ by considering that all the non-retrieved documents are not relevant.

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}$$

Small values of V and V_i

$$P(k_i \mid R) = \frac{V_i}{V}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}$$

a problem when V=1 and V_i=0

- alternative 1

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0.5}{N - V + 1}$$

- alternative 2

$$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}$$

Probabilistic Model

- Q: “gold silver truck”
  D1: “Shipment of gold damaged in a fire”
  D2: “Delivery of silver arrived in a silver truck”
  D3: “Shipment of gold arrived in a truck”
- IDF (Select Keywords)
  - a = in = of = 0 = log $^{3/3}$
    arrived = gold = shipment = truck = 0.176 = log $^{3/2}$
    damaged = delivery = fire = silver = 0.477 = log $^{3/1}$
- 8 Keywords (Dimensions) are selected
  - arrived(1), damaged(2), delivery(3), fire(4), gold(5), silver(6), shipment(7), truck(8)

Probabilistic Model

- Initial Guess

$$P(k_i \mid R) = 0.5$$

$$P(k_i \mid \overline{R}) = \frac{N_i}{N} \quad (N = 3)$$

$$Sim(d_i, q) = \sum_{i=1}^{t} g_i(d_i) \times g_i(q) \times \log\left(\frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}\right) \quad (t = 8)$$

	1	2	3	4	5	6	7	8
N_i	2	1	1	1	2	1	2	2

$$Sim(d_1, q) = \log\left(\frac{0.5 \times \frac{1}{3}}{\frac{2}{3} \times 0.5}\right) = \log\left(\frac{1}{2}\right) = -\log^{2} = -0.30103$$

$$Sim(d_2, q) = 0$$

$$Sim(d_3, q) = -2 \times \log^{2} = -0.60206$$

$$Sim(d_2, q) > Sim(d_1, q) > Sim(d_3, q)$$

Probabilistic Model

- Interaction with User?
  - Relevance Feedback
- How many documents need to be retrieved?

No Interaction with User

- Retrieve 1 Document: d2 (relevant)

$$V = 1 \quad \& \quad N = 3$$

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{N_i - V_i + 0.5}{N - V + 1} \quad (N = 3)$$

$$Sim(d_i, q) = \sum_{i=1}^{t} g_i(d_i) \times g_i(q) \times \log\left(\frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}\right) \quad (t = 8)$$

	1	2	3	4	5	6	7	8
V_i	1	0	1	0	0	1	0	1
N_i	2	1	1	1	2	1	2	2

$$Sim(d_1, q) = \log\left(\frac{\frac{0.5}{2} \times \frac{0.5}{3}}{\frac{2.5}{3} \times \frac{1.5}{2}}\right) = -(\log^{5} + \log^{3}) = -1.17609$$

$$Sim(d_2, q) = 2 \times \log^{3} + \log^{5} = 1.65321$$

$$Sim(d_3, q) = -\log^{5} = -0.69897$$

$$Sim(d_2, q) > Sim(d_3, q) > Sim(d_1, q)$$

No Interaction with User

- Retrieve 2 Documents: d2 (relevant) & d1

$$V = 2 \quad \& \quad N = 3$$

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{N_i - V_i + 0.5}{N - V + 1} \quad (N = 3)$$

$$\mathrm{Sim}(d_i, q) = \sum_{i=1}^{t} g_i(d_i) \times g_i(q) \times \log\left(\frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}\right) \quad (t = 8)$$

	1	2	3	4	5	6	7	8
V_i	1	1	1	1	1	1	1	1
N_i	2	1	1	1	2	1	2	2

$$\mathrm{Sim}(d_1, q) = \log\left(\frac{\frac{0.5}{2} \times \frac{1.5}{3}}{\frac{1.5}{3} \times \frac{1.5}{2}}\right) = -\log^{3} = -0.47712$$

$$\mathrm{Sim}(d_2, q) = 0$$

$$\mathrm{Sim}(d_3, q) = -2 \times \log^{3} = -0.95424$$

$$\mathrm{Sim}(d_2, q) > \mathrm{Sim}(d_1, q) > \mathrm{Sim}(d_3, q)$$

No Interaction with User

- Retrieve 3 Documents: d2, d1 (non-relevant) &d3

$V = 3 \quad \& \quad N = 3 \quad \& \quad V_i = N_i$

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{N_i - V_i + 0.5}{N - V + 1} \quad (N = 3)$$

$$Sim(d_j, q) = \sum_{i=1}^{t} g_i(d_j) \times g_i(q) \times \log\left(\frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}\right) \quad (t = 8)$$

	1	2	3	4	5	6	7	8
V_i	2	1	1	1	2	1	2	2
N_i	2	1	1	1	2	1	2	2

$$Sim(d_1, q) = \log\left(\frac{\frac{0.5}{2} \times \frac{1.5}{3}}{\frac{1.5}{3} \times \frac{1.5}{2}}\right) = -\log{}^{3} = -0.47712$$

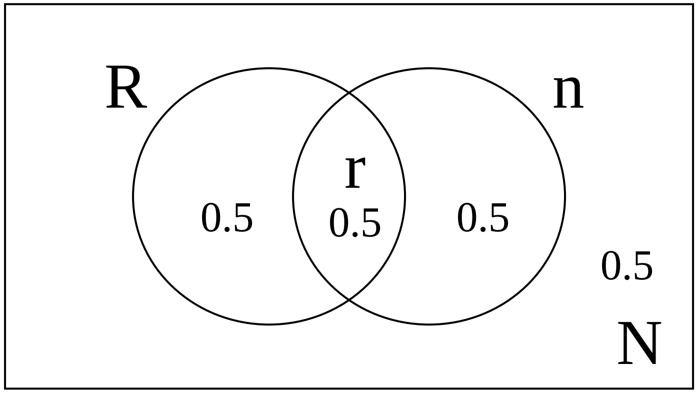
$$Sim(d_2, q) = 0$$

$$Sim(d_3, q) = 2 \times (\log{}^{5} - \log{}^{3}) = 0.44370$$

$Sim(d_3, q) > Sim(d_1, q) > Sim(d_2, q)$ → We need to interact with user.

Interaction with User

- Retrieve 2 Documents: d2 & d1 (non-relevant)



N = # of documents in the collection

n = # of documents indexed by a given term

R = # of relevant documents

r = # of relevant documents indexed by the given term

$$P(k_i \mid R) = \frac{r}{R}$$

$$P(k_i \mid \overline{R}) = \frac{n}{N} \quad (N = 3)$$

$$P(k_i \mid R) = \frac{r + 0.5}{R + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n + 1}{N + 2} \quad (N = 3)$$

$$Sim(d_i, q) = \sum_{i=1}^{t} g_i(d_i) \times g_i(q) \times \log\left(\frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}\right) \quad (t = 8)$$

Interaction with User

- Alternative 2

$$P(k_i \mid R) = \frac{r + 0.5}{R + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n - r + 0.5}{N - R + 1}$$

- Alternative 3

$$P(k_i \mid R) = \frac{r + 0.5}{R - r + 0.5}$$

$$P(k_i \mid \overline{R}) = \frac{n + 1}{N - n + 1}$$

- Alternative 4

$$P(k_i \mid R) = \frac{r + 0.5}{R - r + 0.5}$$

$$P(k_i \mid \overline{R}) = \frac{n - r + 0.5}{(N - n) - (R - r) + 0.5}$$

Interaction with User

	1	2	3	4	5	6	7	8
N	3	3	3	3	3	3	3	3
n	2	1	1	1	2	1	2	2
R	1	1	1	1	1	1	1	1
r	1	0	1	0	0	1	0	1
$P(k_i \mid R) = \frac{r+0.5}{R+1}$	$\frac{1.5}{2}$	$\frac{0.5}{2}$	$\frac{1.5}{2}$	$\frac{0.5}{2}$	$\frac{0.5}{2}$	$\frac{1.5}{2}$	$\frac{0.5}{2}$	$\frac{1.5}{2}$
$P(k_i \mid \overline{R}) = \frac{n+1}{N+2}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{3}{5}$

$$\mathrm{Sim}(d_1, q) = \log\left(\frac{\frac{0.5}{2} \times \frac{2}{5}}{\frac{3}{5} \times \frac{1.5}{2}}\right) = \log\frac{2}{9} = -0.65321$$

$$\mathrm{Sim}(d_2, q) = \log\left(\frac{\frac{1.5}{2} \times \frac{3}{5}}{\frac{2}{5} \times \frac{0.5}{2}}\right) + \log\left(\frac{\frac{1.5}{2} \times \frac{2}{5}}{\frac{3}{5} \times \frac{0.5}{2}}\right) = \log^{9} = 0.95424$$

$$\mathrm{Sim}(d_3, q) = \log\left(\frac{\frac{0.5}{2} \times \frac{2}{5}}{\frac{3}{5} \times \frac{1.5}{2}}\right) + \log\left(\frac{\frac{1.5}{2} \times \frac{2}{5}}{\frac{3}{5} \times \frac{0.5}{2}}\right) = \log\frac{4}{9} = -0.35218$$

$$\mathrm{Sim}(d_2, q) > \mathrm{Sim}(d_3, q) > \mathrm{Sim}(d_1, q)$$

Analysis of Probabilistic Model

- advantage
  - documents are ranked in decreasing order of their probability of being relevant
- disadvantages
  - the need to guess the initial separation of documents into relevant and non-relevant sets
  - do not consider the frequency with which an index terms occurs inside a document
  - the independence assumption for index terms

Comparison of classic models

- Boolean model: the weakest classic model
- Vector model is expected to outperform the probabilistic model with general collections (Salton and Buckley)

Okapi at TREC3 and TREC4

SE Robertson, S Walker, S Jones, MM Hancock-Beaulieu, M Gatford

Department of Information Science

City University

$$sim(d_j, q) \approx \frac{P(\overrightarrow{d_j} \mid R)}{P(\overrightarrow{d_j} \mid \overline{R})}$$

$$\approx \sum_{i=1}^{t} g_i(\overrightarrow{d_j}) g_i(\vec{q}) \times \log \frac{P(k_i \mid R) \times (1 - P(k_i \mid \overline{R}))}{P(k_i \mid \overline{R}) \times (1 - P(k_i \mid R))}$$

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1} \qquad 1 - P(k_i \mid R) = 1 - \frac{V_i + 0.5}{V + 1} = \frac{V - V_i + 0.5}{V + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0.5}{N - V + 1} \qquad 1 - P(k_i \mid \overline{R}) = 1 - \frac{n_i - V_i + 0.5}{N - V + 1} = \frac{N - V - n_i + V_i + 0.5}{N - V + 1}$$

$$sim(d_j, q) \approx \log \frac{\dfrac{V_i + 0.5}{V + 1} \times \dfrac{N - V - n_i + V_i + 0.5}{N - V + 1}}{\dfrac{n_i - V_i + 0.5}{N - V + 1} \times \dfrac{V - V_i + 0.5}{V + 1}}$$

$$= \log \frac{(V_i + 0.5) \times (N - V - n_i + V_i + 0.5)}{(n_i - V_i + 0.5) \times (V - V_i + 0.5)}$$

BM25 function in Okapi

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2 \, |Q| \, \frac{avdl-dl}{avdl+dl}$$

term frequency and document length

used for long query

$$\frac{(k_2+1)qtf}{k_2+qtf}$$

Q: a query, containing terms T

$w^{(1)}$: Robertson-Sparck Jones weight $log \frac{(r+0.5)\times(N-n-R+r+0.5)}{(n-r+0.5)\times(R-r+0.5)}$

N: the number of documents in the collection (note: N)

n: the number of documents containing the term (note: n_i)

R: the number of documents known to be relevant to a specific topic (note: V)

r: the number of relevant documents containing the term (note: V_i)

K: $k_1((1-b)+b*dl/avdl)$ $k_1=0$: binary model (no term frequency); k_1=large value (using raw term frequency); b=1 (fully scaling the term weight by document length); b=0 (no length normalization)

k_1, b, k_2 and k_3: parameters depend on the database and nature of topics in TREC4 experiments, k_1, k_3 and b were 1.0-2.0, 8 and 0.6-0.75, respectively., and k_2 was zero throughout

tf: frequency of occurrence of the term within a specific document (note: k_i)

qtf: the frequency of the term within the topic from which Q was derived

dl: document length

avdl: average document length

Fuzzy Set Model

Alternative Set Theoretic Models -Fuzzy Set Model

- Model
  - a query term: a fuzzy set
  - a document: degree of membership in this set
  - membership function
    - Associate membership function with the elements of the class
    - 0: no membership in the set
    - 1: full membership
    - 0~1: marginal elements of the set

documents

Fuzzy Set Theory

a class

- A fuzzy subset A of a universe of discourse U is characterized by a membership function μ_A: U→[0,1] which associates with each element u of U a number $\mu_A(u)$ in the interval [0,1]

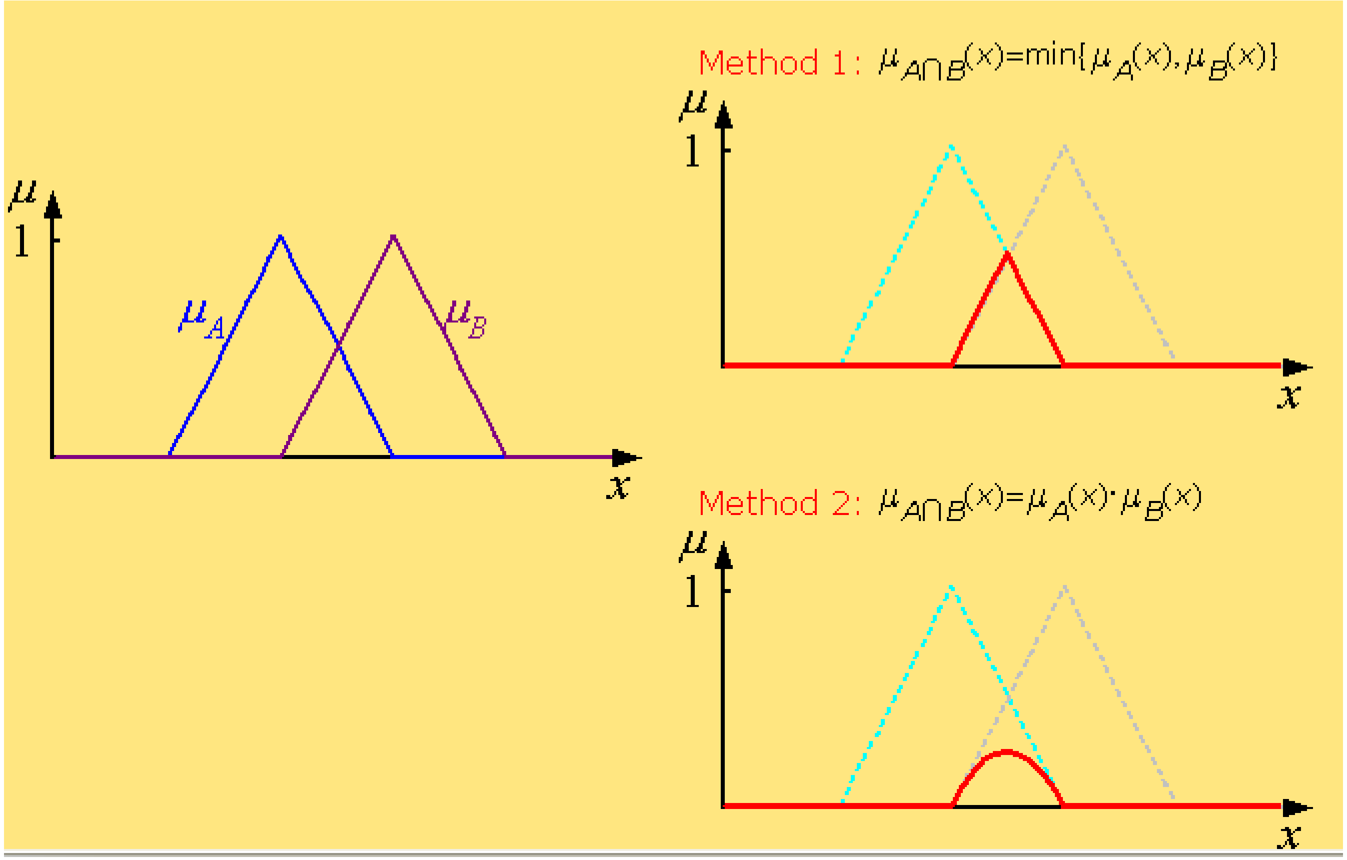
a document

  - complement: $\mu_{\overline{A}}(u) = 1 - \mu_A(u)$
  - union: $\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$
  - intersection: $\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$

Examples

- Assume U={d_1, d_2, d_3, d_4, d_5, d_6}
- Let A and B be {d_1, d_2, d_3} and {d_2, d_3, d_4}, respectively.
- Assume μ_A={d_1:0.8, d_2:0.7, d_3:0.6, d_4:0, d_5:0, d_6:0} and μ_B={d_1:0, d_2:0.6, d_3:0.8, d_4:0.9, d_5:0, d_6:0}
- $\mu_{\overline{A}}(u) = 1 - \mu_A(u)$ ={d_1:0.2, d_2:0.3, d_3:0.4, d_4:1, d_5:1, d_6:1}
- $\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$={$d_1$:0.8, d_2:0.7, d_3:0.8, d_4:0.9, d_5:0, d_6:0}
- $\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$={$d_1$:0, d_2:0.6, d_3:0.6, d_4:0, d_5:0, d_6:0}

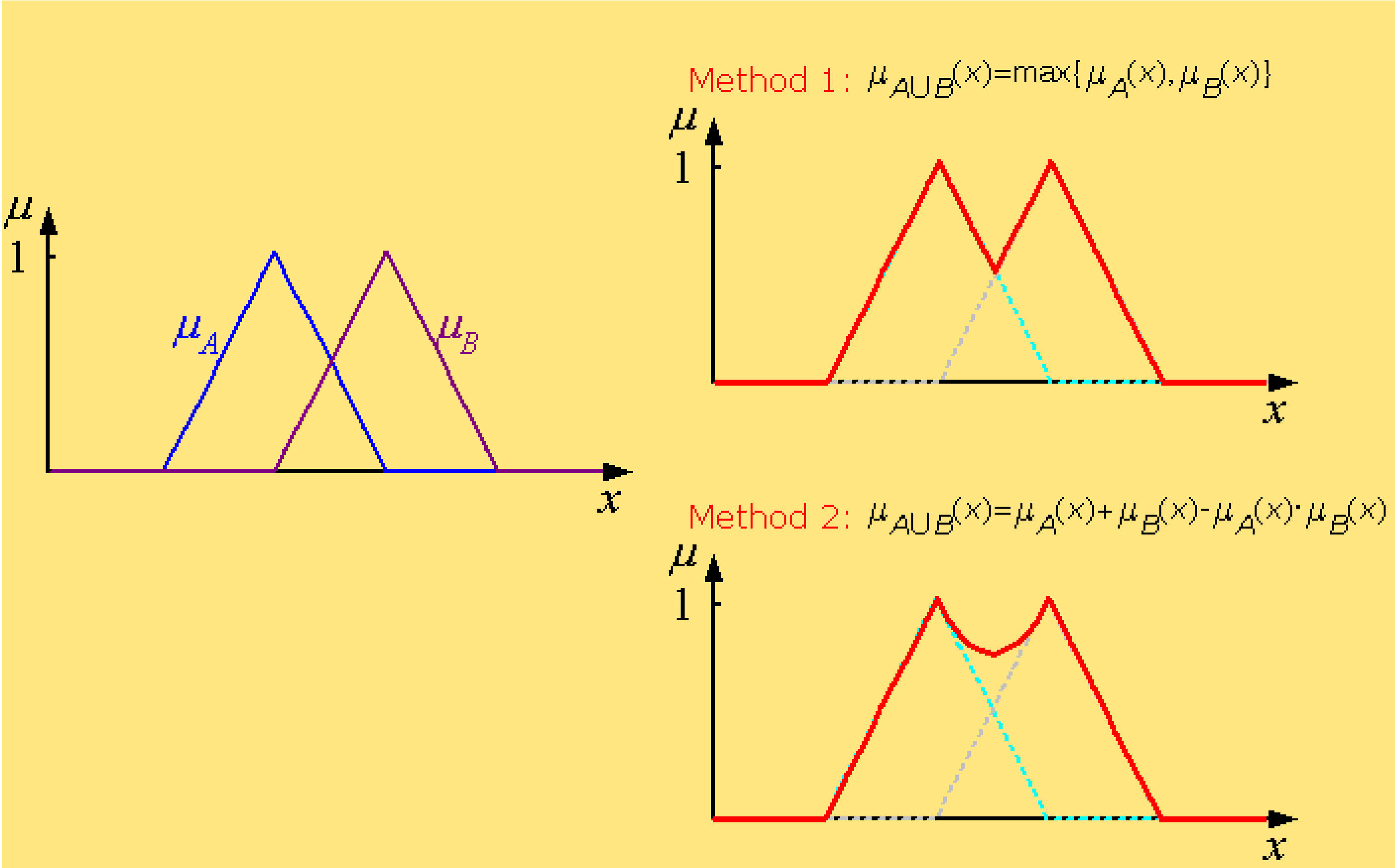
Fuzzy AND



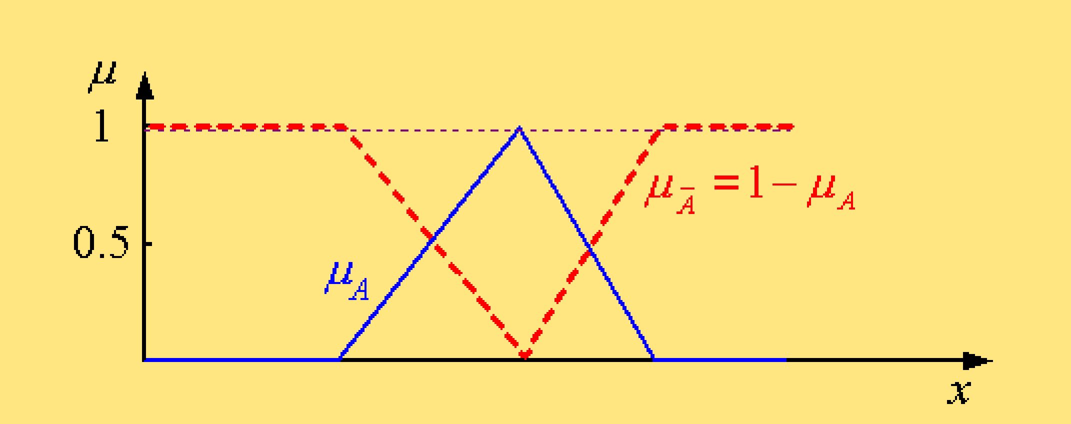
Method 1: $\mu_{A\cap B}(x)=\min\{\mu_A(x),\mu_B(x)\}$

Method 2: $\mu_{A\cap B}(x)=\mu_A(x)\cdot\mu_B(x)$

Fuzzy OR


μ
1
μ_A
μ_B
x
Method 1: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$
μ
1
x
Method 2: $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$
μ
1
x

Fuzzy NOT



Fuzzy Information Retrieval

- basic idea
  - Expand the set of index terms in the query with related terms (from the thesaurus) such that additional relevant documents can be retrieved
  - A thesaurus can be constructed by defining a term-term correlation matrix $\vec{c}$ whose rows and columns are associated to the index terms in the document collection

keyword connection matrix

Fuzzy Information Retrieval

(Continued)

- normalized correlation factor $c_{i,l}$ between two terms k_i and k_l (0~1)

$$c_{i,l} = \frac{n_{i,l}}{n_i + n_l - n_{i,l}} \quad \text{where} \begin{cases} n_i \text{ is \# of documents containing term } k_i \\ n_l \text{ is \# of documents containing term } k_l \\ n_{i,l} \text{ is \# of documents containing } k_i \text{ and } k_l \end{cases}$$

- In the fuzzy set associated to each index term k_i, a document d_j has a degree of membership $\mu_{i,j}$

$$\mu_{i,j} = 1 - \prod_{k_l \in d_j} (1 - c_{i,l})$$

Fuzzy Information Retrieval

(Continued)

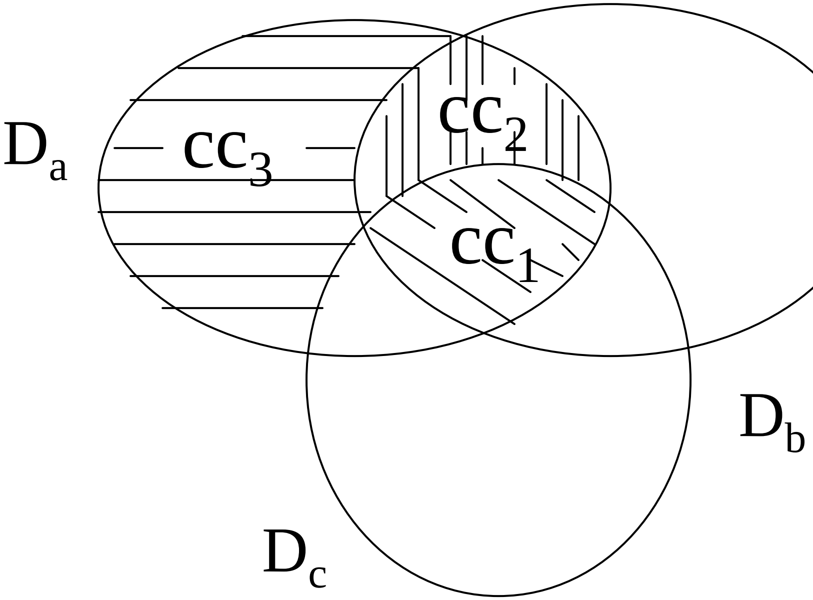
- physical meaning
  - A document d_j belongs to the fuzzy set associated to the term k_i if its own terms are related to k_i, i.e., $\mu_{i,j}$=1.
  - If there is at least one index term k_l of d_j which is strongly related to the index k_i, then $\mu_{i,j}$~1.

    k_i is a good fuzzy index
  - When all index terms of d_j are only loosely related to k_i, $\mu_{i,j}$~0.

    k_i is not a good fuzzy index

Example

- $q=(k_a \wedge (k_b \vee \neg k_c)$
  $=(k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \neg k_c) \vee (k_a \wedge \neg k_b \wedge \neg k_c)$
  $=cc_1+cc_2+cc_3$



D_a: the fuzzy set of documents associated to the index k_a

$d_j \in D_a$ has a degree of membership $\mu_{a,j}$ > a predefined threshold K

$\overline{D_a}$: the fuzzy set of documents associated to the index $\overline{k_a}$ (the negation of index term k_a)

Example

Query q=$k_a \wedge (k_b \vee \neg\, k_c)$

disjunctive normal form $\vec{q}_{dnf}$=(1,1,1) ∨ (1,1,0) ∨ (1,0,0)

(1) the degree of membership in a disjunctive fuzzy set is computed using an algebraic sum *(instead of max function*) *more smoothly*
(2) the degree of membership in a conjunctive fuzzy set is computed using an algebraic product (*instead of min function*) *more smoothly*

$$\boxed{\text{Recall} \quad \mu_{\overline{A}}(u) = 1 - \mu_A(u)}$$

$$\begin{aligned}\mu_{q,j} &= \mu_{cc1+cc2+cc3,j} \\ &= 1 - \prod_{i=1}^{3}(1-\mu_{cc_i,j}) \\ &= 1-(1-\mu_{a,j}\mu_{b,j}\mu_{c,j})\times(1-\mu_{a,j}\mu_{b,j}(1-\mu_{c,j}))\times(1-\mu_{a,j}(1-\mu_{b,j})(1-\mu_{c,j}))\end{aligned}$$

Fuzzy Set Model

- Q: “gold silver truck”
  D1: “Shipment of gold damaged in a fire”
  D2: “Delivery of silver arrived in a silver truck”
  D3: “Shipment of gold arrived in a truck”
- IDF (Select Keywords)
  - a = in = of = 0 = log $^{3/3}$
    arrived = gold = shipment = truck = 0.176 = log $^{3/2}$
    damaged = delivery = fire = silver = 0.477 = log $^{3/1}$
- 8 Keywords (Dimensions) are selected
  - arrived(1), damaged(2), delivery(3), fire(4), gold(5), silver(6), shipment(7), truck(8)

Fuzzy Set Model

$$
\begin{aligned}
\mu_{gold,d1} &= 1 - \prod_{k_1 \in d_1} (1 - C_{gold,k1}) \\
&= 1 - (1 - C_{gold,shipment}) * (1 - C_{gold,gold}) * (1 - C_{gold,damaged}) * (1 - C_{gold,fire}) \\
&= 1 - (1 - \frac{2}{2+2-2}) * (1 - \frac{1}{2+1-1}) * (1 - \frac{2}{2+2-2}) * (1 - \frac{2}{2+1-1}) \\
&= 1 - 0 * \frac{1}{2} * 0 * \frac{1}{2} \\
&= 1
\end{aligned}
$$

$$
\mu_{silver,d1} = 1 - 1 * 1 * 1 * 1 = 0
$$

$$
\begin{aligned}
\mu_{truck,d1} &= 1 - \prod_{k_1 \in d_1} (1 - C_{truck,k1}) \\
&= 1 - (1 - C_{truck,shipment}) * (1 - C_{truck,gold}) * (1 - C_{truck,damaged}) * (1 - C_{truck,fire}) \\
&= 1 - (1 - \frac{1}{2+2-1}) * (1 - \frac{1}{2+2-1}) * (1 - \frac{0}{2+1-0}) * (1 - \frac{0}{2+1-0}) \\
&= 1 - \frac{2}{3} * \frac{2}{3} * 1 * 1 \\
&= \frac{5}{9}
\end{aligned}
$$

Fuzzy Set Model

$$\mu_{gold,d2} = 1 - 1 * 1 * \frac{2}{3} * \frac{2}{3} = \frac{5}{9}$$

$$\mu_{silver,d2} = 1$$

$$\mu_{truck,d2} = 1$$

$$\mu_{gold,d3} = 1$$

$$\mu_{silver,d3} = 1 - 1 * 1 * \frac{1}{2} * \frac{1}{2} = \frac{3}{4}$$

$$\mu_{truck,d3} = 1$$

Fuzzy Set Model

- Sim(q,d): Alternative 1

$$\mu_{q,d1} = \mu_{gold\wedge silver\wedge truck,d1} = \mu_{gold,d1} * \mu_{silver,d1} * \mu_{truck,d1} = 0$$

$$\mu_{q,d2} = \mu_{gold\wedge silver\wedge truck,d1} = \mu_{gold,d2} * \mu_{silver,d2} * \mu_{truck,d2} = \frac{5}{9}$$

$$\mu_{q,d3} = \mu_{gold\wedge silver\wedge truck,d1} = \mu_{gold,d3} * \mu_{silver,d3} * \mu_{truck,d3} = \frac{3}{4}$$

Sim(q,d_3) > Sim(q,d_2) > Sim(q,d_1)

- Sim(q,d): Alternative 2

$$\mu_{q,d1} = \mu_{gold\wedge silver\wedge truck,d1} = \min(\mu_{gold,d1}, \mu_{silver,d1}, \mu_{truck,d1}) = 0$$

$$\mu_{q,d2} = \mu_{gold\wedge silver\wedge truck,d1} = \min(\mu_{gold,d2}, \mu_{silver,d2}, \mu_{truck,d2}) = \frac{5}{9}$$

$$\mu_{q,d3} = \mu_{gold\wedge silver\wedge truck,d1} = \min(\mu_{gold,d3}, \mu_{silver,d3}, \mu_{truck,d3}) = \frac{3}{4}$$

Sim(q,d_3) > Sim(q,d_2) > Sim(q,d_1)

Generalized Vector Space Model

Alternative Algebraic Model: Generalized Vector Space Model

- independence of index terms
  - $\vec{k_i}$: a vector associated with the index term k_i
  - the set of vectors $\{\vec{k_1}, \vec{k_2}, \ldots, \vec{k_t}\}$ is linearly independent
    - orthogonal: $\vec{k}_i \bullet \vec{k}_j = 0$ for i≠j
    - **Theorem:** If the nonzero vectors **k**1, **k**2, · · · , **k***n* are orthogonal, then they are linearly independent.
  - The index term vectors are assumed linearly independent but are not pairwise orthogonal in generalized vector space model
  - The index term vectors, which are not seen as the basis of the space, are composed of *smaller components* derived from the particular collection.

Review

- Two vectors u and v are linearly independent
  - if $\alpha u+\beta v=0$ then $\alpha=\beta=0$
- Two vectors u and v are orthogonal, I.e, $\theta=90^o$
  - $u \bullet v=0$ (I.e., $u^Tv=0$)
- if two vectors u and v are orthogonal, then u and v are linearly independent
  - assume $\alpha u+\beta v=0$, $u\neq0$ and $v\neq0$
  - $u^T(\alpha u+\beta v)=0$ --> $\alpha\ u^Tu+\beta\ u^T\ v=0$ --> $\alpha u^Tu=0$

Generalized Vector Space Model

- $\{k_1, k_2, \ldots, k_t\}$: index terms in a collection
- $w_{i,j}$: binary weights associated with the term-document pair $\{k_i, d_j\}$
- The patterns of term *co-occurrence* (inside documents) can be represented by a set of 2^t *minterms*

m_1=(0, 0, …, 0): point to documents containing none of index terms

m_2=(1, 0, …, 0): point to documents containing the index term k_1 only

m_3=(0,1,…,0): point to documents containing the index term k_2 only

m_4=(1,1,…,0): point to documents containing the index terms k_1 and k_2

…

m_{2^t}=(1, 1, …, 1): point to documents containing all the index terms

- $g_i(m_j)$: return the weight {0,1} of the index term k_i in the minterm m_j $(1 \le i \le t)$

Generalized Vector Space Model

(*Continued*)

$$\vec{m}_1 = (1,0,\ldots,0,0)$$

$$\vec{m}_2 = (0,1,\ldots,0,0)$$

$$\ldots$$

$$\vec{m}_{2^t} = (0,0,\ldots,0,1)$$

$$\vec{m}_i \bullet \vec{m}_j = 0 \ \ for\ i \neq j$$

(the set of $\vec{m}_i$ are pairwise orthogonal)

- $\vec{m}_i$ (2^t-tuple vector) is associated with minterm m_i (t-tuple vector)
- e.g., $\vec{m}_4$ is associated with m_4 containing k_1 and k_2, and no others
- co-occurrence of index terms inside documents: dependencies among index terms

t=3

minterm m_r	$\vec{m}_r$ vector	d1 (k1)	d11 (k1 k2)
$m_1=(0,0,0)$	$\vec{m}_1=(1,0,0,0,0,0,0,0)$	d2 (k3)	d12 (k1 k3)
$m_2=(0,0,1)$	$\vec{m}_2=(0,1,0,0,0,0,0,0)$	d3 (k3)	d13 (k1 k2)
$m_3=(0,1,0)$	$\vec{m}_3=(0,0,1,0,0,0,0,0)$	d4 (k1)	d14 (k1 k2)
$m_4=(0,1,1)$	$\vec{m}_4=(0,0,0,1,0,0,0,0)$	d5 (k2)	d15 (k1 k2 k3)
$m_5=(1,0,0)$	$\vec{m}_5=(0,0,0,0,1,0,0,0)$	d6 (k2)	d16 (k1 k2)
$m_6=(1,0,1)$	$\vec{m}_6=(0,0,0,0,0,1,0,0)$	d7 (k2 k3)	d17 (k1 k2)
$m_7=(1,1,0)$	$\vec{m}_7=(0,0,0,0,0,0,1,0)$	d8 (k2 k3)	d18 (k1 k2)
$m_8=(1,1,1)$	$\vec{m}_8=(0,0,0,0,0,0,0,1)$	d9 (k2)	d19 (k1 k2 k3)
		d10 (k2 k3)	d20 (k1 k2)

$$\vec{k}_1 = \frac{c_{1,5}\vec{m}_5 + c_{1,6}\vec{m}_6 + c_{1,7}\vec{m}_7 + c_{1,8}\vec{m}_8}{\sqrt{{c_{1,5}}^2 + {c_{1,6}}^2 + {c_{1,7}}^2 + {c_{1,8}}^2}}$$

$$c_{1,5} = w_{1,1} + w_{1,4} \quad c_{1,6} = w_{1,12}$$

$$c_{1,7} = w_{1,11} + w_{1,13} + w_{1,14} + w_{1,16} + w_{1,17} + w_{1,18} + w_{1,20}$$

$$c_{1,8} = w_{1,15} + w_{1,19}$$

t=3

minterm m_r	$\vec{m}_r$ vector	d1 (k1)	d11 (k1 k2)
m_1=(0,0,0)	$\vec{m}_1$=(1,0,0,0,0,0,0,0)	d2 (k3)	d12 (k1 k3)
m_2=(0,0,1)	$\vec{m}_2$=(0,1,0,0,0,0,0,0)	d3 (k3)	d13 (k1 k2)
m_3=(0,1,0)	$\vec{m}_3$=(0,0,1,0,0,0,0,0)	d4 (k1)	d14 (k1 k2)
m_4=(0,1,1)	$\vec{m}_4$=(0,0,0,1,0,0,0,0)	d5 (k2)	d15 (k1 k2 k3)
m_5=(1,0,0)	$\vec{m}_5$=(0,0,0,0,1,0,0,0)	d6 (k2)	d16 (k1 k2)
m_6=(1,0,1)	$\vec{m}_6$=(0,0,0,0,0,1,0,0)	d7 (k2 k3)	d17 (k1 k2)
m_7=(1,1,0)	$\vec{m}_7$=(0,0,0,0,0,0,1,0)	d8 (k2 k3)	d18 (k1 k2)
m_8=(1,1,1)	$\vec{m}_8$=(0,0,0,0,0,0,0,1)	d9 (k2)	d19 (k1 k2 k3)
		d10 (k2 k3)	d20 (k1 k2)

$$\vec{k}_2 = \frac{c_{2,3}\vec{m}_3 + c_{2,4}\vec{m}4 + c_{2,7}\vec{m}_7 + c_{2,8}\vec{m}_8}{\sqrt{{c_{2,3}}^2 + {c_{2,4}}^2 + {c_{2,7}}^2 + {c_{2,8}}^2}}$$

$$c_{2,3} = w_{2,5} + w_{2,6} + w_{2,9} \qquad c_{2,4} = w_{2,7} + w_{2,8} + w_{2,10}$$

$$c_{2,7} = w_{2,11} + w_{2,13} + w_{2,14} + w_{2,16} + w_{2,17} + w_{2,18} + w_{2,20}$$

$$c_{2,8} = w_{2,15} + w_{2,19}$$

	minterm m_r	$\vec{m}_r$ vector	d1 (k1)	d11 (k1 k2)
	$m_1=(0,0,0)$	$\vec{m}_1=(1,0,0,0,0,0,0,0)$	d2 (k3)	d12 (k1 k3)
	$m_2=(0,0,1)$	$\vec{m}_2=(0,1,0,0,0,0,0,0)$	d3 (k3)	d13 (k1 k2)
	$m_3=(0,1,0)$	$\vec{m}_3=(0,0,1,0,0,0,0,0)$	d4 (k1)	d14 (k1 k2)
t=3	$m_4=(0,1,1)$	$\vec{m}_4=(0,0,0,1,0,0,0,0)$	d5 (k2)	d15 (k1 k2 k3)
	$m_5=(1,0,0)$	$\vec{m}_5=(0,0,0,0,1,0,0,0)$	d6 (k2)	d16 (k1 k2)
	$m_6=(1,0,1)$	$\vec{m}_6=(0,0,0,0,0,1,0,0)$	d7 (k2 k3)	d17 (k1 k2)
	$m_7=(1,1,0)$	$\vec{m}_7=(0,0,0,0,0,0,1,0)$	d8 (k2 k3)	d18 (k1 k2)
	$m_8=(1,1,1)$	$\vec{m}_8=(0,0,0,0,0,0,0,1)$	d9 (k2)	d19 (k1 k2 k3)
			d10 (k2 k3)	d20 (k1 k2)

$$\vec{k}_3 = \frac{c_{3,2}\vec{m}_2 + c_{3,4}\vec{m}4 + c_{3,6}\vec{m}_6 + c_{3,8}\vec{m}_8}{\sqrt{{c_{3,2}}^2 + {c_{3,4}}^2 + {c_{3,6}}^2 + {c_{3,8}}^2}}$$

$$c_{3,2} = w_{3,2} + w_{3,3} \qquad c_{3,4} = w_{3,7} + w_{3,8} + w_{3,10} \qquad c_{3,6} = w_{3,12}$$

$$c_{3,8} = w_{3,15} + w_{3,19}$$

Generalized Vector Space Model

(*Continued*)

- Determine the index vector $\overrightarrow{k_i}$ associated with the index term k_i

$$\vec{k}_i = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \overrightarrow{m}_r}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c^2_{i,r}}}$$

Collect all the vectors $\vec{m}_r$ in which the index term k_i is in state 1.

$$c_{i,r} = \sum_{d_j | g_l(\vec{d}_j) = g_l(m_r) \text{ for all } l} w_{i,j}$$

Sum up $w_{i,j}$ associated with the index term k_i and document d_j whose term occurrence pattern coincides with minterm m_r

Generalized Vector Space Model

(*Continued*)

- $\overrightarrow{k_i} \bullet \overrightarrow{k_j}$ quantifies a degree of correlation between k_i and k_j

$$\vec{k}_i \bullet \vec{k}_j = \sum_{\forall r | g_i(m_r)=1 \wedge g_j(m_r)=1} c_{i,r} \times c_{j,r}$$

- standard cosine similarity is adopted

$$\vec{d}_j = \sum\nolimits_{\forall i} w_{i,j} \vec{k}_i \qquad \vec{q} = \sum\nolimits_{\forall i} w_{i,q} \vec{k}_i$$

$$\vec{k}_i = \frac{\sum_{\forall r, g_i(m_r)=1} c_{i,r} \overrightarrow{m_r}}{\sqrt{\sum_{\forall r, g_i(m_r)=1} c^2_{i,r}}}$$

$$\vec{k}_1 = \frac{c_{1,5}\vec{m}_5 + c_{1,6}\vec{m}6 + c_{1,7}\vec{m}_7 + c_{1,8}\vec{m}_8}{\sqrt{{c_{1,5}}^2 + {c_{1,6}}^2 + {c_{1,7}}^2 + {c_{1,8}}^2}}$$

$$\vec{k}_2 = \frac{c_{2,3}\vec{m}_3 + c_{2,4}\vec{m}4 + c_{2,7}\vec{m}_7 + c_{2,8}\vec{m}_8}{\sqrt{{c_{2,3}}^2 + {c_{2,4}}^2 + {c_{2,7}}^2 + {c_{2,8}}^2}}$$

$$\vec{k}_3 = \frac{c_{3,2}\vec{m}_2 + c_{3,4}\vec{m}4 + c_{3,6}\vec{m}_6 + c_{3,8}\vec{m}_8}{\sqrt{{c_{3,2}}^2 + {c_{3,4}}^2 + {c_{3,6}}^2 + {c_{3,8}}^2}}$$

$$\vec{k}_1 \bullet \vec{k}_2 = (c_{1,7} \times c_{2,7} + c_{1,8} \times c_{2,8}) /$$

$$(\sqrt{{c_{1,5}}^2 + {c_{1,6}}^2 + {c_{1,7}}^2 + {c_{1,8}}^2} \times \sqrt{{c_{2,3}}^2 + {c_{2,4}}^2 + {c_{2,7}}^2 + {c_{2,8}}^2})$$

$$\vec{k}_1 \bullet \vec{k}_3 = (c_{1,6} \times c_{3,6} + c_{1,8} \times c_{3,8}) / \ldots$$

$$\vec{k}_2 \bullet \vec{k}_3 = (c_{2,4} \times c_{3,4} + c_{2,8} \times c_{3,8}) / \ldots$$

Latent Semantic Indexing Model

Vector Space Model: Pros

- **Automatic** selection of index terms
- **Partial matching** of queries and documents *(dealing with the case where no document contains all search terms)*
- **Ranking** according to **similarity score** *(dealing with large result sets)*
- **Term weighting** schemes *(improves retrieval performance)*
- Various extensions
  - Document clustering
  - Relevance feedback (modifying query vector)

Problems with Lexical Semantics

- Ambiguity and association in natural language
  - **Polysemy**: Words often have a **multitude of meanings** and different types of usage *(more severe in very heterogeneous collections).*
  - The vector space model is unable to discriminate between different meanings of the same word.

$$\mathrm{sim}_{\mathtt{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

Problems with Lexical Semantics

- **Synonymy**: Different terms may have an **dentical or a similar meaning** (weaker: words indicating the same topic).
- No associations between words are made in the vector space representation.

$$\mathrm{sim}_{\mathtt{true}}(d, q) > \cos(\angle(\vec{d}, \vec{q}))$$

Latent Semantic Indexing (LSI) Model

- representation of documents and queries by index terms
  - problem 1: many unrelated documents might be included in the answer set
  - problem 2: relevant documents which are not indexed by any of the query keywords are not retrieved
- possible solution: concept matching instead of index term matching
  - application in cross-language information retrieval (CLIR)

basic idea

- Map each document and query vector into a lower dimensional space which is associated with concepts
- Retrieval in the reduced space may be superior to retrieval in the space of index terms

Definition

- t: the number of index terms in the collection
- N: the total number of documents
- $\vec{M}=(M_{ij})$: a term-document association matrix with t rows (i.e., term) and N columns (i.e., document)
- M_{ij}: a weight $w_{i,j}$ associated with the term-document pair $[k_i, d_j]$ (e.g., using tf-idf)

Singular Value Decomposition

$A \in R^{n\times n}$

(1) $A = A^t$

$\exists Q \in R^{n\times n} \quad st\ QQ^t = I \quad \{Q^tQ = I\}$ orthogonal

sin*gular value decomposition*:

$A = QDQ^t \quad \{A^t = (QDQ^t)^t = (Q^t)^t D^t Q^t = QDQ^t = A\}$

$$\text{where D} = \begin{pmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \cdot & & \\ & 0 & & \cdot & \\ & & & & \cdot \quad \lambda_n \end{pmatrix} \quad \text{diagonal matrix}$$

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$

$A \in R^{n \times n}$

(2) $A \neq A^t$

$\exists U, V \in R^{n \times n} \quad st \; U^t U = I, V^t V = I$ orthogonal

$\sin gular \; value \; decomposition:$

$$\boxed{(AB)^T = B^T A^T}$$

$A = UDV^t$

$AA^t = (UDV^t)(UDV^t)^t = (UDV^t)(VDU^t) = UD^2U^t$

where D = $\begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}$ diagonal matrix

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$

$$A = QDQ^t$$

$$AQ = QDQ^tQ = QD$$

$$where\ Q = [q_1 \quad q_2 \quad \dots \quad q_n], \quad q_i : a\ column\ vector$$

$$A[q_1 \quad q_2 \dots q_n] = [q_1 \quad q_2 \dots q_n] \begin{pmatrix} \lambda_1 & & & & & 0 \\ & \lambda_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \lambda_n \end{pmatrix}$$

$$[Aq_1\ Aq_2 \dots Aq_n] = [\lambda_1 q_1 \quad \lambda_2 q_2 \quad \dots \lambda_n q_n]$$

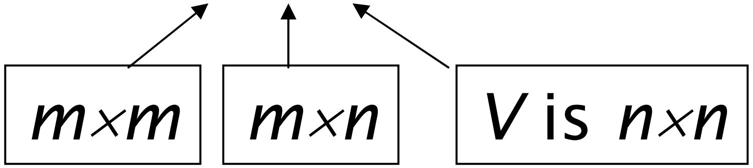
$$Aq_1 = \lambda_1 q_1 \quad Aq_2 = \lambda_2 q_2 \quad \dots \quad Aq_n = \lambda_n q_n$$

$\lambda_1, \lambda_2, \dots, \lambda_n$ 為A之eigenvalues，
q_k為A相對於λ_k之eigenvector

Singular Value Decomposition

For an $m \times n$ matrix $\mathbf{A}$ of rank r there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

$$A = U \Sigma V^T$$



$m \times m$

$m \times n$

V is $n \times n$

The columns of ***U*** are orthogonal eigenvectors of $\boldsymbol{AA^T}$.

The columns of ***V*** are orthogonal eigenvectors of $\boldsymbol{A^TA}$.

Eigenvalues $\lambda_1 \ldots \lambda_r$ of $\boldsymbol{AA^T}$ are the eigenvalues of $\boldsymbol{A^TA}$.

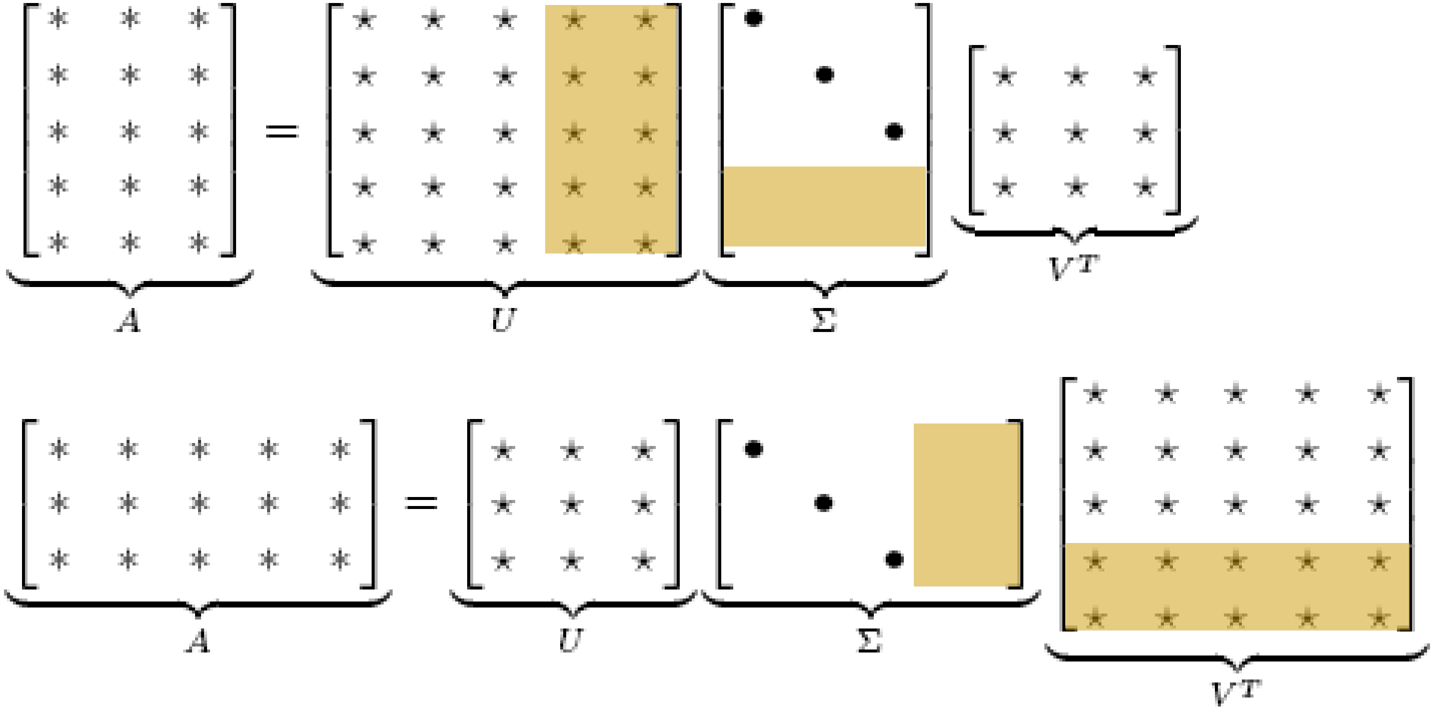
$$\sigma_i = \sqrt{\lambda_i}$$

$$\Sigma = diag\left(\sigma_1 \ldots \sigma_r\right)$$

Singular values.

Singular Value Decomposition

- Illustration of SVD dimensions and



SVD example

Let $A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

Thus m=3, n=2. Its SVD is

$$\begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Typically, the singular values arranged in decreasing order.

Singular Value Decomposition

$$\overrightarrow{M} : a\ term - document\ matrix\ with\ t\ rows\ and\ N\ columns$$

$$\overrightarrow{M} = \overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t$$

$$\overrightarrow{M}^t \overrightarrow{M} : a\ N \times N\ document - to - document\ matrix$$

$$\overrightarrow{M}\,\overrightarrow{M}^t : a\ t \times t\ term - to - term\ matrix$$

According to

$$\overrightarrow{M} \in R^{t \times N}$$

$$\exists \overrightarrow{K} : the\ matrix\ of\ eigenvectors\ derived\ from\ \overrightarrow{M}\,\overrightarrow{M}^t \quad \overrightarrow{K}^t \overrightarrow{K} = I$$

$$\overrightarrow{D} : the\ matrix\ of\ eigenvectors\ derived\ from\ \overrightarrow{M}^t \overrightarrow{M} \quad \overrightarrow{D}^t \overrightarrow{D} = I$$

$$\overrightarrow{M} = \overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t$$

$$\overrightarrow{M}^t\,\overrightarrow{M} : document-to-document\ matrix$$

$$= (\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)^t(\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)$$

$$= (\overrightarrow{D}\overrightarrow{S}^t\overrightarrow{K}^t)(\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)$$

$$= \overrightarrow{D}\overrightarrow{S}^2\overrightarrow{D}^t$$

$$\overrightarrow{M}\,\overrightarrow{M}^t : term-to-term\ matrix$$

$$= (\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)(\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)^t$$

$$= (\overrightarrow{K}\overrightarrow{S}\overrightarrow{D}^t)(\overrightarrow{D}\overrightarrow{S}^t\overrightarrow{K}^t)$$

$$= \overrightarrow{K}\overrightarrow{S}^2\overrightarrow{K}^t$$

對照A=QDQ^t

Q is matrix of eigenvectors of A

D is diagonal matrix of singular values

得到

$\overrightarrow{D}$: *the matrix of eigenvectors*

derived from $\overrightarrow{M}^t\,\overrightarrow{M}$

$\overrightarrow{K}$: *the matrix of eigenvectors*

derived from $\overrightarrow{M}\,\overrightarrow{M}^t$

$\overrightarrow{S} : r \times r$ *diagonal matrix of* sin*gular*

values, where $r = \min(t, N)$

s < r (Concept space is reduced)

Consider only the s largest singular values of $\vec{S}$

$$\begin{pmatrix} \lambda_1 & & & & \\ & \lambda_2 & & 0 & \\ & & \cdot & & \\ 0 & & & \cdot & \\ & & & \cdot & \lambda_n \end{pmatrix}$$

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$$

The resultant $\vec{M}_s$ matrix is the matrix of rank s which is closest to the original matrix M in the least square sense.

$$\overrightarrow{M}_s = \overrightarrow{K}_s \overrightarrow{S}_s \overrightarrow{D}_s^t$$

(s<<t, s<<N)

由概念分群來說明：
太細-各個index term代表不同的概念
太粗-所有index term成為一概念

s必須足夠大到涵蓋所有相關文件，
也不能太粗，把不相關的納進來。

Latent Semantic Indexing (LSI)

- Perform a **low-rank approximation** of **document-term matrix** (typical rank **100-300**)
- General idea
  - Map documents (*and* terms) to a **low-dimensional** representation.
  - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
  - Compute document similarity based on the **inner product** in this **latent semantic space**

Goals of LSI

- Similar terms map to similar location in low dimensional space
- Noise reduction by dimension reduction

What it is

- From term-doc matrix A, we compute the approximation $A_{k.}$
- There is a row for each term and a column for each doc in A_k
- Thus docs live in a space of $k<<r$ dimensions
  - These dimensions are not the original axes

Ranking in LSI

- query: a pseudo-document in the original $\vec{\text{M}}$ term-document
  - query is modeled as the document with number 0
  - $\vec{\text{M}}_s{}^t\vec{\text{M}}_s$: the ranks of all documents w.r.t this query

$$\overrightarrow{M}_s{}^t\overrightarrow{M}_s = (\overrightarrow{K}_s\overrightarrow{S}_s\overrightarrow{D}_s^t)^t\overrightarrow{K}_s\overrightarrow{S}_s\overrightarrow{D}_s^t$$

$$= \overrightarrow{D}_s\overrightarrow{S}_s\overrightarrow{K}_s^t\overrightarrow{K}_s\overrightarrow{S}_s\overrightarrow{D}_s^t = \overrightarrow{D}_s\overrightarrow{S}_s\overrightarrow{S}_s\overrightarrow{D}_s^t$$

$$= (\overrightarrow{D}_s\overrightarrow{S}_s)(\overrightarrow{D}_s\overrightarrow{S}_s)^t$$

(i,j) qualifies the relationship between documents d_i and d_j When i = 0, it denotes similarity between q and documents

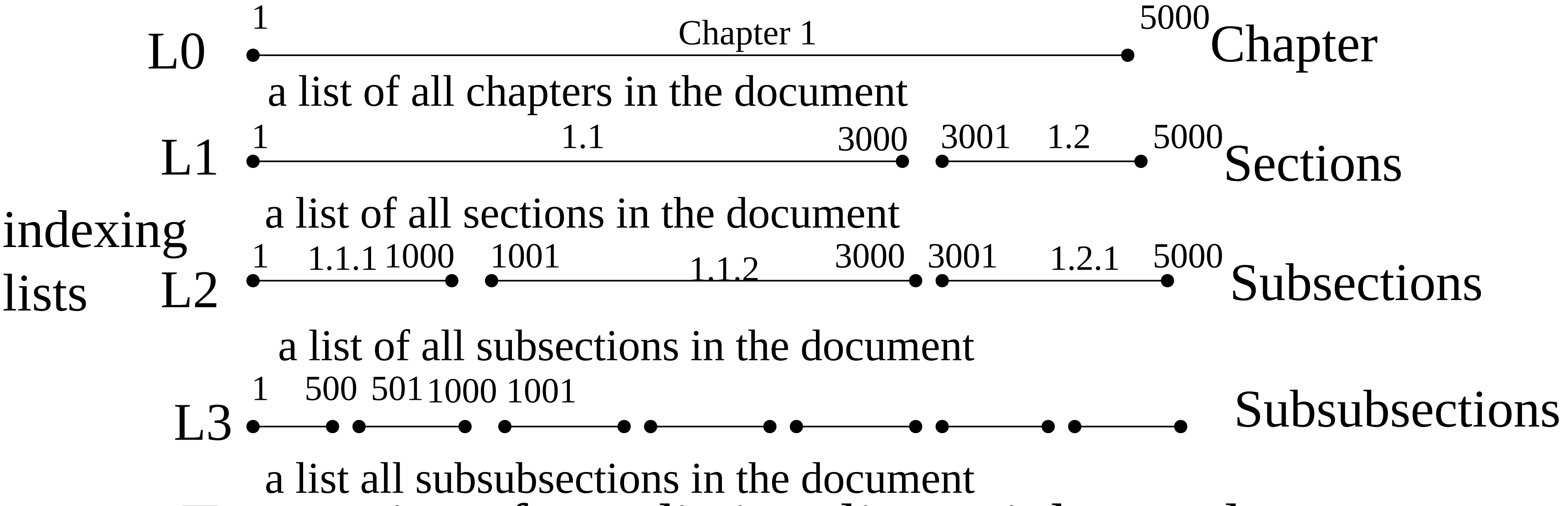
Structured Text Retrieval Models

- Definition
  - Combine information on text content with information on the document structure
  - e.g., same-page(near('atomic holocaust', Figure(label('earth'))))
- Expressive power vs. evaluation efficiency
  - a model based on *non-overlapping lists*
  - a model based on *proximal nodes*
- Terminology
  - match point: position in the text of a sequence of words that matches the user query
  - region: a contiguous portion of the text
  - node: a structural component of the document (chap, sec, …)

Non-Overlapping Lists

- divide the whole text of each document in non-overlapping text regions (*lists*)
- example

non-overlapping in a list



L0 1 Chapter 1 5000 Chapter

a list of all chapters in the document

L1 1 1.1 3000 3001 1.2 5000 Sections

a list of all sections in the document

indexing lists

L2 1 1.1.1 1000 1001 1.1.2 3000 3001 1.2.1 5000 Subsections

a list of all subsections in the document

L3 1 500 501 1000 1001 Subsubsections

a list all subsubsections in the document

- Text regions from distinct lists might overlap

Non-Overlapping Lists

(*Continued*)

Recall that there is another inverted file for the words in the text

- Data structure
  - – a single inverted file
  - – each structural component (e.g., chap, sec, …) stands as an entry
  - – for each entry, there is a list of text regions as a list occurrences
- Operations
  - – Select a region which contains a given word
  - – Select a region A which does not contain any other region B (where B belongs to a list distinct from the list for A)
  - – Select a region not contained within any other region
  - – …

Inverted Files

- File is represented as an array of indexed records.

	Term 1	Term 2	Term 3	Term 4
Record 1	1	1	0	1
Record 2	0	1	1	1
Record 3	1	0	1	1
Record 4	0	0	1	1

Inverted-file process

- The record-term array is inverted (transposed).

	Record 1	Record 2	Record 3	Record 4
Term 1	1	0	1	0
Term 2	1	1	0	0
Term 3	0	1	1	1
Term 4	1	1	1	1

Inverted-file process (*Continued*)

- Take two or more rows of an inverted term-record array, and produce a single combined list of record identifiers.

Query (term2 and term3)

1 1 0 0

0 1 1 1

----

1 <-- R2

Extensions of Inverted Index Operations (Distance Constraints)

- Distance Constraints
  - (A within sentence B)
    terms A and B must co-occur in a common sentence
  - (A adjacent B)
    terms A and B must occur adjacently in the text

Extensions of Inverted Index Operations (Distance Constraints)

- Implementation
  - include term-location in the inverted indexes
    information: {R345, R348, R350, …}
    retrieval: {R123, R128, R345, …}
  - include sentence-location in the indexes
    information:
    {R345, 25; R345, 37; R348, 10; R350, 8; …}
    retrieval:
    {R123, 5; R128, 25; R345, 37; R345, 40; …}

Extensions of Inverted Index Operations (Distance Constraints)

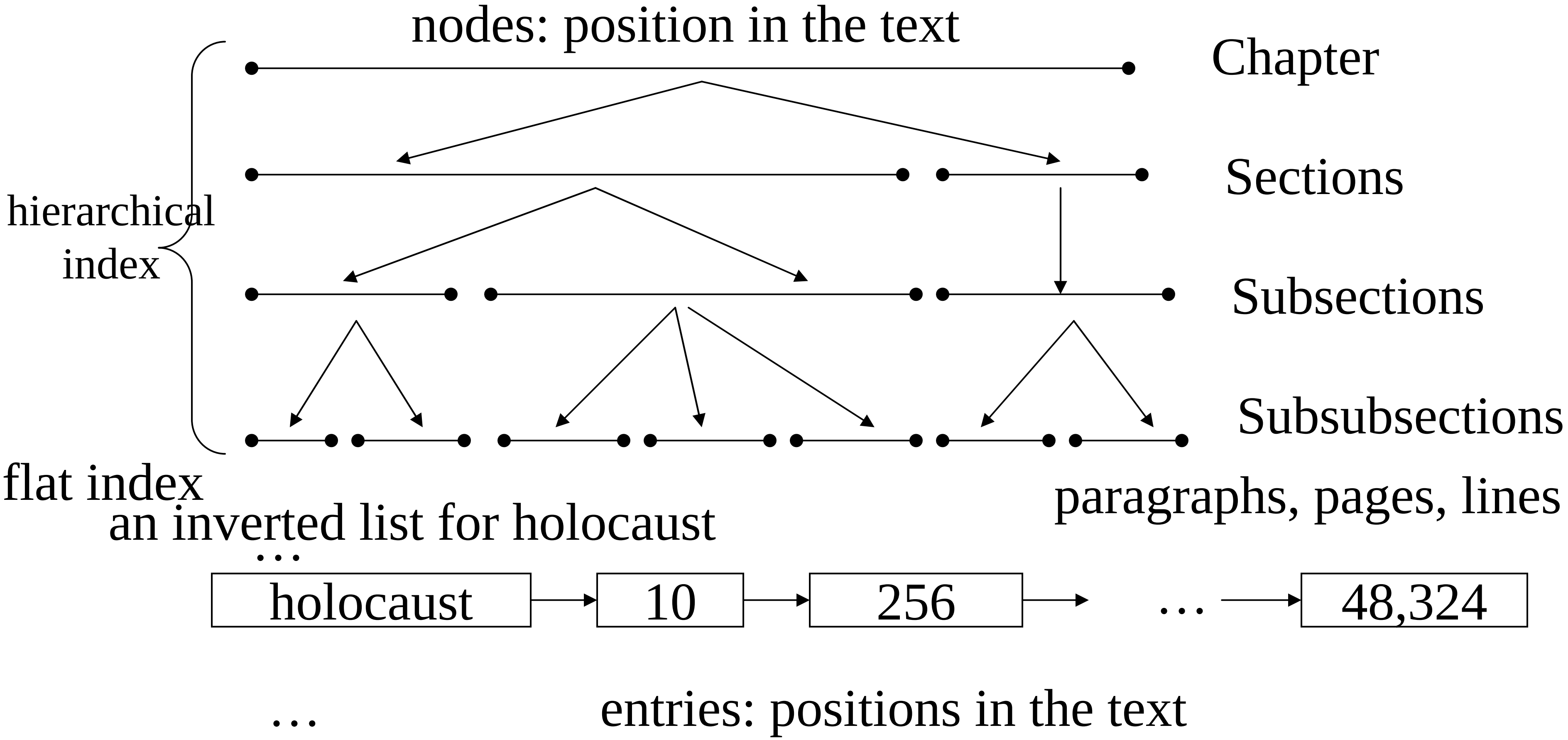
– include paragraph numbers in the indexes
  sentence numbers within paragraphs
  word numbers within sentences
  information: {R345, 2, 3, 5; …}
  retrieval: {R345, 2, 3, 6; …}

– query examples
  (information adjacent retrieval)
  (information within five words retrieval)

– cost: the size of indexes

Model Based on Proximal Nodes

- hierarchical vs. flat indexing structures



nodes: position in the text

Chapter

Sections

hierarchical index

Subsections

Subsubsections

flat index

paragraphs, pages, lines

an inverted list for holocaust

…

holocaust → 10 → 256 → … → 48,324

…

entries: positions in the text

Model Based on Proximal Nodes

(*Continued*)

- query language
  - – Specification of regular expressions
  - – Reference to structural components by name
  - – Combination
  - – Example
    - Search for sections, subsections, or subsubsections which contain the word ‘holocaust’
    - [(*section) with (‘holocaust’)]

Model Based on Proximal Nodes

(*Continued*)

- Basic algorithm
  - Traverse the inverted list for the term ‘holocaust’
  - For each entry in the list (i.e., an occurrence), search the hierarchical index looking for sections, subsections, and sub-subsections
- Revised algorithm
  - For the first entry, search as before
  - Let the last matching structural component be the innermost matching component
  - Verify the innermost matching component also matches the second entry.
    - If it does, the larger structural components above it also do.

nearby nodes

Models for Browsing

- Browsing vs. searching
  - The goal of a searching task is clearer in the mind of the user than the goal of a browsing task
- Models
  - Flat browsing
  - Structure guided browsing
  - The hypertext model

Models for Browsing

- Flat organization
  - Documents are represented as dots in a 2-D plan
  - Documents are represented as elements in a 1-D list, e.g., the results of search engine
- Structure guided browsing
  - Documents are organized in a directory, which group documents covering related topics
- Hypertext model
  - Navigating the hypertext: a traversal of a directed graph

Trends and Research Issues

- Library systems
  - Cognitive and behavioral issues oriented particularly at a better understanding of which criteria the users adopt to judge relevance
- Specialized retrieval systems
  - e.g., legal and business documents
  - how to retrieve all relevant documents without retrieving a large number of unrelated documents
- The Web
  - User does not know what he wants or has great difficulty in formulating his request
  - How the paradigm adopted for the user interface affects the ranking
  - The indexes maintained by various Web search engine are almost disjoint