Models for Information Retrieval

- Classical IR Models

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. *Modern Information Retrieval*, Chapter 2
2. *Language Modeling for Information Retrieval*, Chapter 3

Index Terms

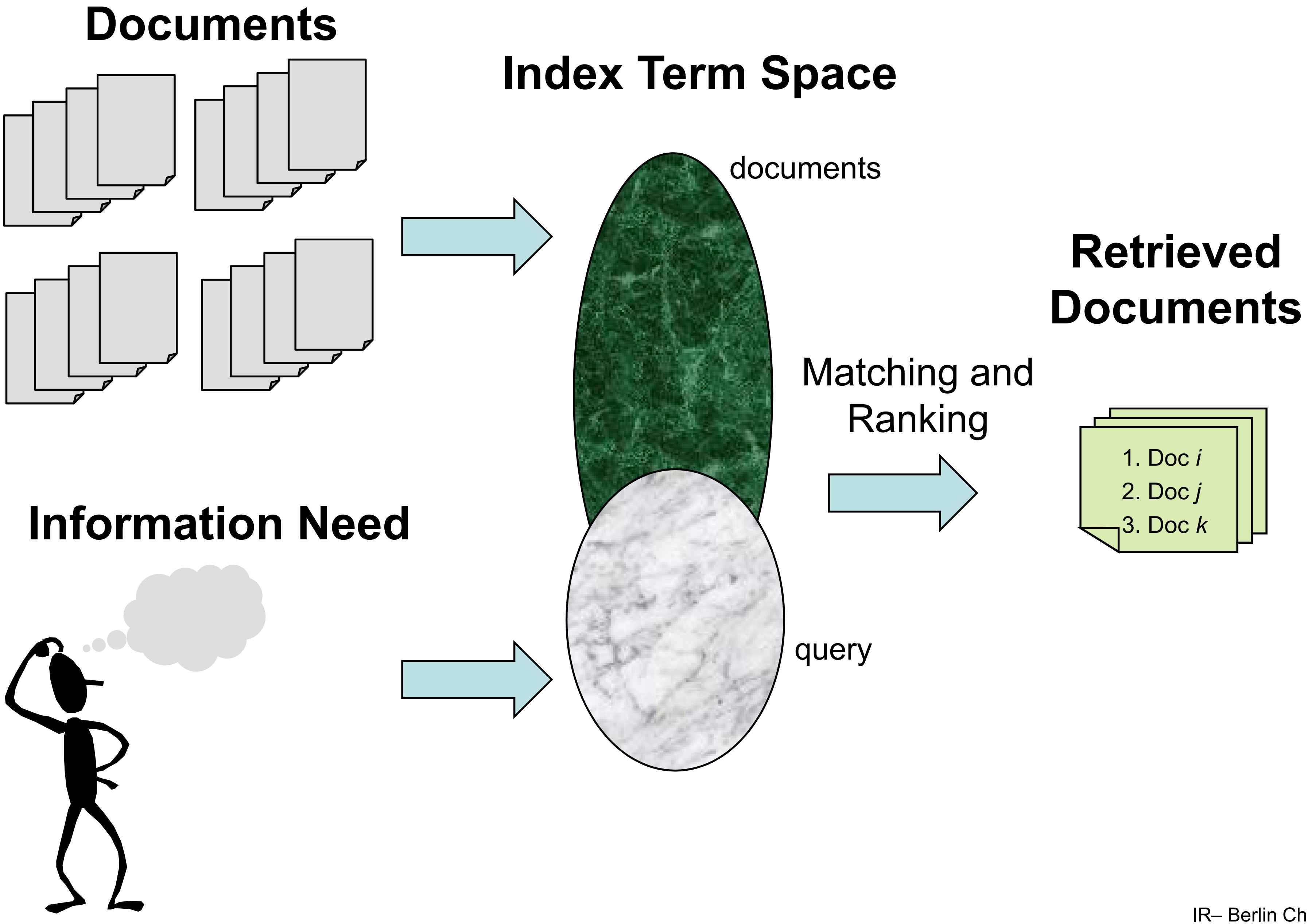
- Meanings From Two Perspectives
  - In a *restricted* sense (keyword-based)
    - An index term is a (predefined) keyword (usually a noun) which has some semantic meaning of its own
  - In a *more general* sense (word-based)
    - An index term is simply any word which appears in the text of a document in the collection
    - Full-text

Index Terms (cont.)

- The semantics (main themes) of the documents and of the user information need should be expressed through sets of index terms
  - Semantics is often lost when expressed through sets of words (e.g., possible, probable, likely)
  - Match between the documents and user queries is in the (imprecise?) space of index terms

Index Terms (cont.)

- Documents retrieved are flrequently irrelevant
  - Since most users have no training in query formation, problem is even worst
    - Not familar with the underlying IR process
    - E.g: frequent dissatisfaction of Web users
  - Issue of deciding document relevance, i.e. ranking, is critical for IR systems



Documents

Index Term Space

documents

Retrieved Documents

Matching and Ranking

1. Doc *i*
2. Doc *j*
3. Doc *k*

Information Need

query

Ranking Algorithms

- Also called the "information retrieval models"
- Ranking Algorithms
  - Predict which documents are relevant and which are not
  - Attempt to establish a simple ordering of the document retrieved
  - Documents at the top of the ordering are more likely to be relevant
  - The core of information retrieval systems

Ranking Algorithms (cont.)

- A ranking is based on fundamental premises regarding the notion of relevance, such as:
  - Common sets of index terms
  - Sharing of weighted terms
  - Likelihood of relevance

literal-term matching

$$P(Q|D) \text{ or } P(Q,D) \text{ ?}$$

  - Sharing of same aspects/concepts

Concept/semantic matching

- Distinct sets of premises lead to a distinct IR models

Ranking Algorithms (cont.)

- Concept Matching vs. Literal Matching

Spoken Query

relevant ?

Transcript of Spoken Document

香港星島日報篇報導引述軍事觀察家的話表示，到二零零五年台灣將完全喪失空中優勢，原因是中國大陸戰機不論是數量或是性能上都將超越台灣，報導指出中國在大量引進俄羅斯先進武器的同時也得加快研發自製武器系統，目前西安飛機製造廠任職的改進型飛豹戰機即將部署尚未與蘇愷三十通道地對地攻擊住宅飛機，以督促遇到挫折的監控其戰機目前也已經取得了重大階段性的認知成果。根據日本媒體報導在台海戰爭隨時可能爆發情況之下北京方面的基本方針，使用高科技答應局部戰爭。因此，解放軍打算在二零零四年前又有包括蘇愷三十二期在內的兩百架蘇霍伊戰鬥機。

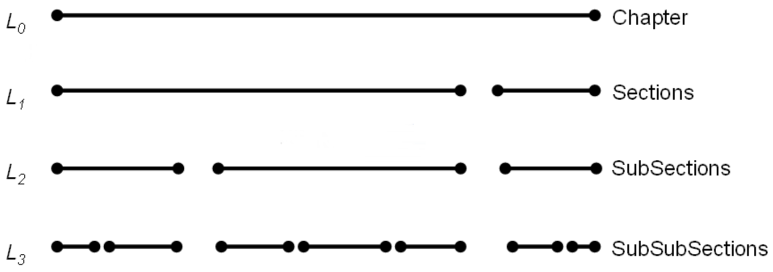
Taxonomy of Classic IR Models

- References to the text content
  - Boolean Model (Set Theoretic)
    - Documents and queries are represented as sets of index terms
  - Vector (Space) Model (Algebraic)
    - Documents and queries are represented as vectors in a *t*-dimensional space
  - Probabilistic Model (Probabilistic)
    - Document and query are represented based on probability theory

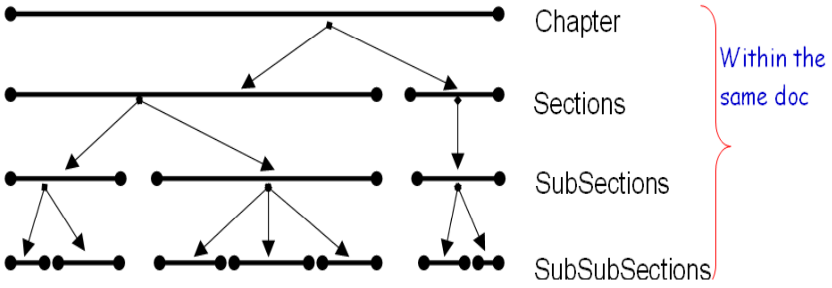
Alternative modeling paradigms will also be extensively studied !

Taxonomy of Classic IR Models (cont.)

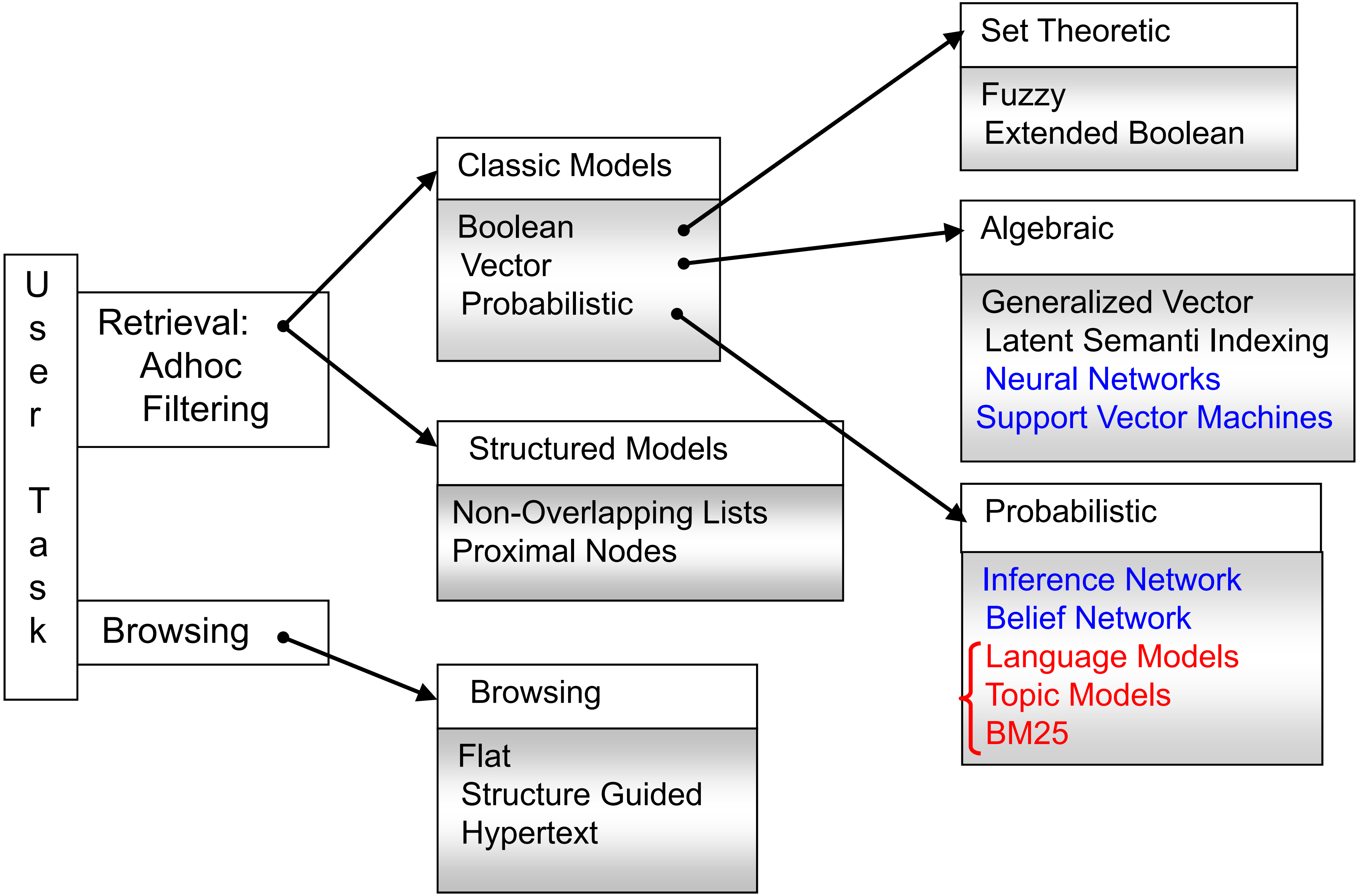
- References to the text structure
  - Non-overlapping list
    - A document divided in *non-overlapping* text regions and represented as multiple lists for chapters, sections, subsections, etc.



  - Proximal Nodes
    - Define a strict hierarchical index over the text which composed of chapters, sections, subsections, paragraphs or lines



Taxonomy of Classic IR Models (cont.)



User Task

Retrieval:
Adhoc
Filtering

Browsing

Classic Models
- Boolean
- Vector
- Probabilistic

Structured Models
- Non-Overlapping Lists
- Proximal Nodes

Browsing
- Flat
- Structure Guided
- Hypertext

Set Theoretic
- Fuzzy
- Extended Boolean

Algebraic
- Generalized Vector
- Latent Semanti Indexing
- Neural Networks
- Support Vector Machines

Probabilistic
- Inference Network
- Belief Network
- Language Models
- Topic Models
- BM25

Taxonomy of Classic IR Models (cont.)

- Three-dimensional Representation

LOGICAL VIEW OF DOCUMENTS

USER TASK

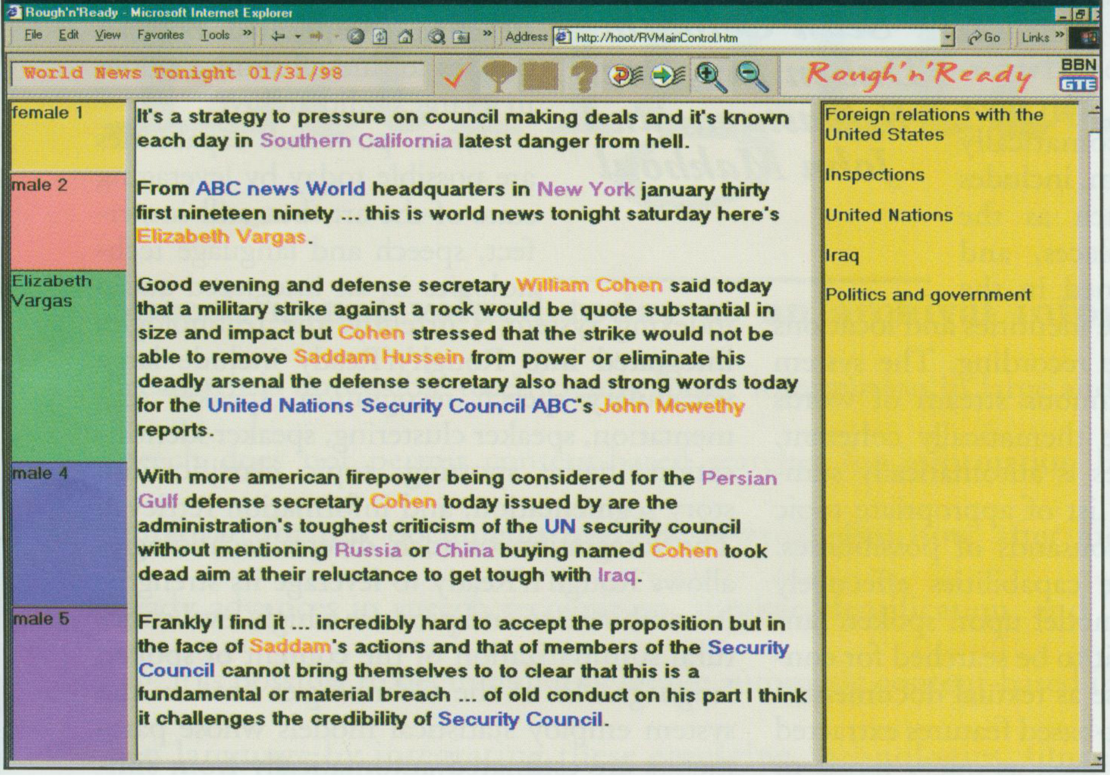
	Index Terms	**Full Text**	**Full Text + Structure**
Retrieval	Classic Set Theoretic Algebraic Probabilistic	Classic Set Theoretic Algebraic Probabilistic	Structured
Browsing	Flat	Flat Hypertext	Structure Guided Hypertext

- The same IR models can be used with distinct document logical views

Browsing the Text Content

- Flat/Structure Guided/Hypertext
- Example (Spoken Document Retrieval)

Figure 1. Elements of the automatic structural summarization produced by Rough'n'Ready.



female 1

It's a strategy to pressure on council making deals and it's known each day in Southern California latest danger from hell.

male 2

From ABC news World headquarters in New York january thirty first nineteen ninety ... this is world news tonight saturday here's Elizabeth Vargas.

Elizabeth Vargas

Good evening and defense secretary William Cohen said today that a military strike against a rock would be quote substantial in size and impact but Cohen stressed that the strike would not be able to remove Saddam Hussein from power or eliminate his deadly arsenal the defense secretary also had strong words today for the United Nations Security Council ABC's John Mcwethy reports.

male 4

With more american firepower being considered for the Persian Gulf defense secretary Cohen today issued by are the administration's toughest criticism of the UN security council without mentioning Russia or China buying named Cohen took dead aim at their reluctance to get tough with Iraq.

male 5

Frankly I find it ... incredibly hard to accept the proposition but in the face of Saddam's actions and that of members of the Security Council cannot bring themselves to to clear that this is a fundamental or material breach ... of old conduct on his part I think it challenges the credibility of Security Council.

Foreign relations with the United States

Inspections

United Nations

Iraq

Politics and government

Browsing the Text Content (cont.)

- Example (Spoken Document Retrieval)

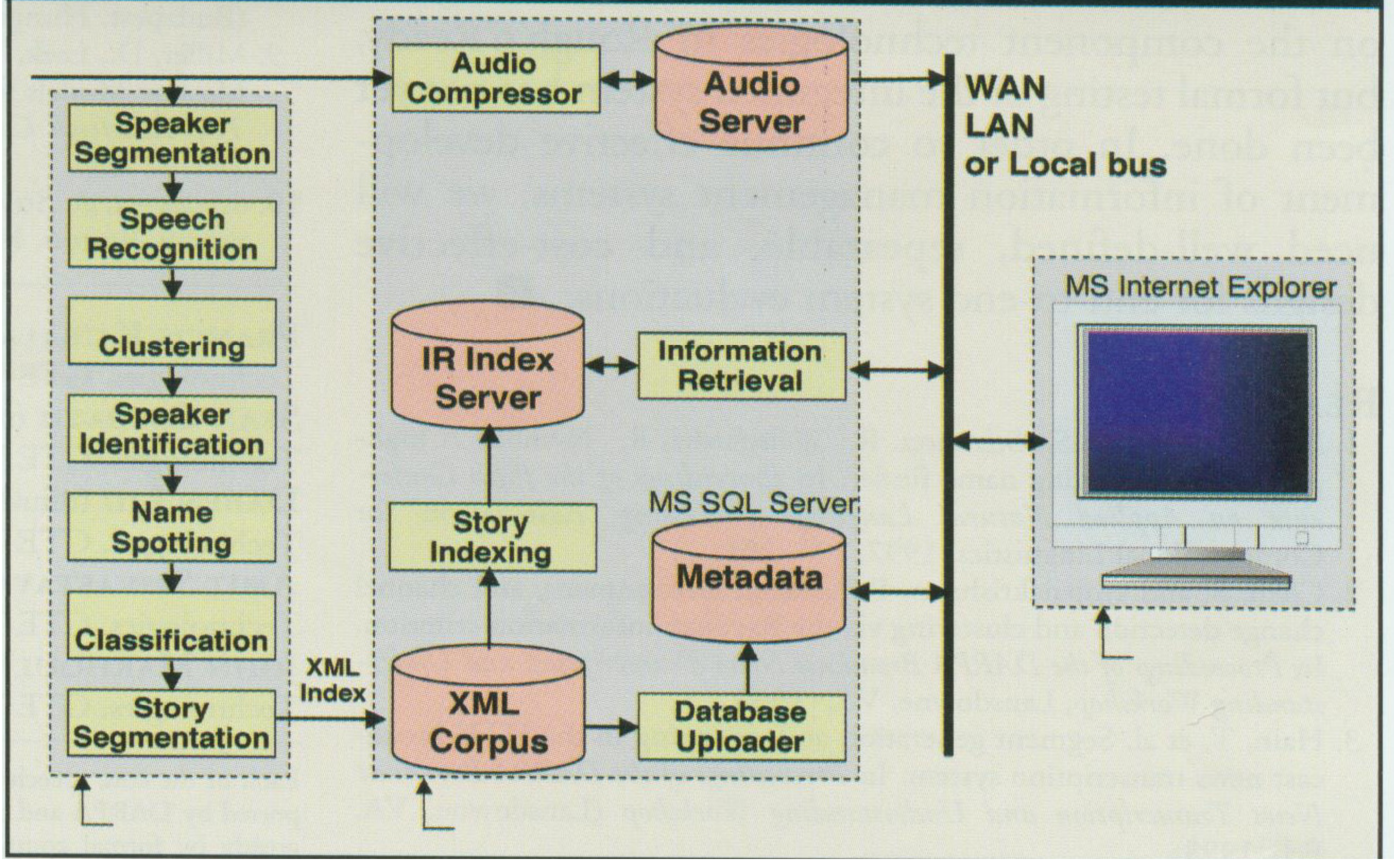
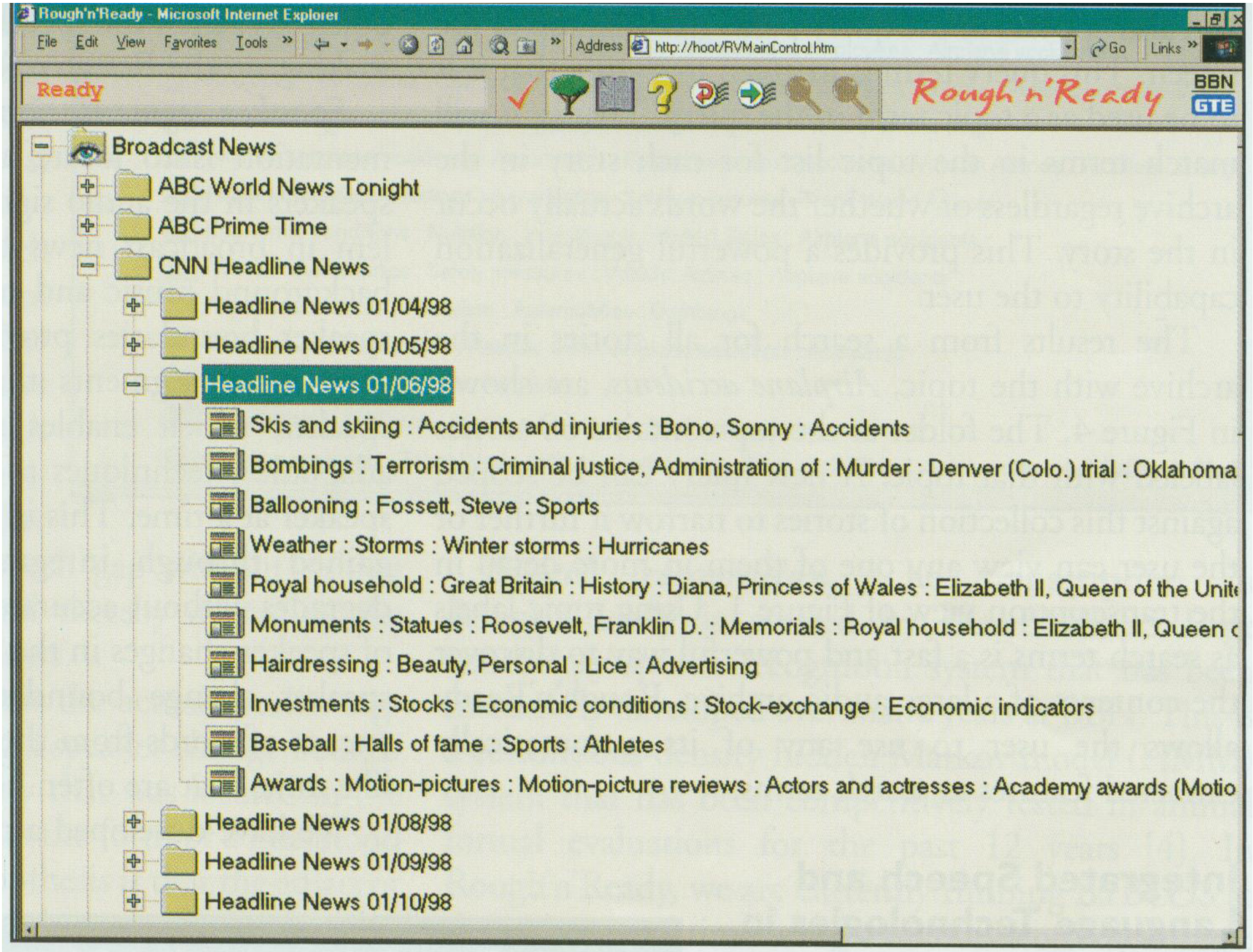


Figure 5. Distinguished architecture of the Rough'n'Ready audio indexing system.

Browsing the Text Content (cont.)

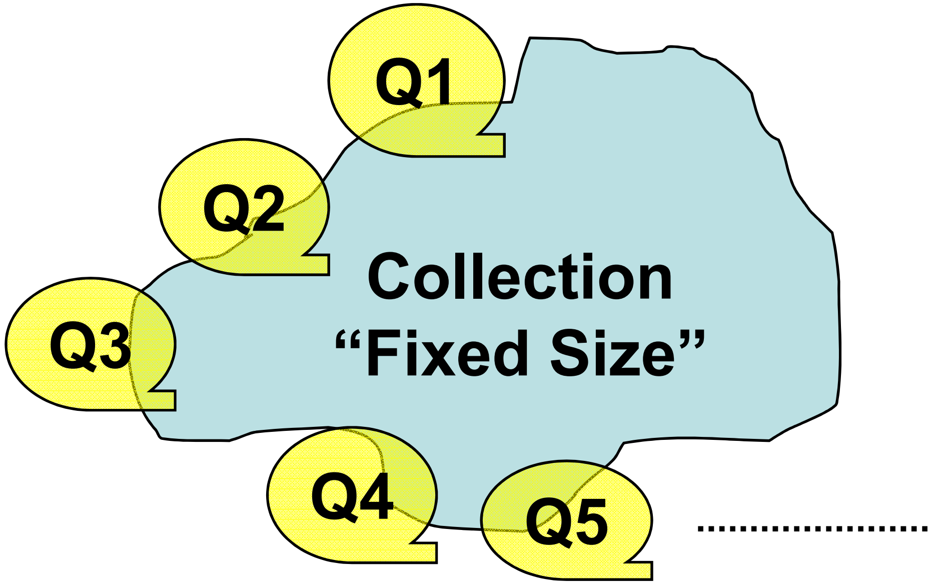
- Example (Spoken Document Retrieval)



- Broadcast News
  - ABC World News Tonight
  - ABC Prime Time
  - CNN Headline News
    - Headline News 01/04/98
    - Headline News 01/05/98
    - Headline News 01/06/98
      - Skis and skiing : Accidents and injuries : Bono, Sonny : Accidents
      - Bombings : Terrorism : Criminal justice, Administration of : Murder : Denver (Colo.) trial : Oklahoma
      - Ballooning : Fossett, Steve : Sports
      - Weather : Storms : Winter storms : Hurricanes
      - Royal household : Great Britain : History : Diana, Princess of Wales : Elizabeth II, Queen of the Unit
      - Monuments : Statues : Roosevelt, Franklin D. : Memorials : Royal household : Elizabeth II, Queen c
      - Hairdressing : Beauty, Personal : Lice : Advertising
      - Investments : Stocks : Economic conditions : Stock-exchange : Economic indicators
      - Baseball : Halls of fame : Sports : Athletes
      - Awards : Motion-pictures : Motion-picture reviews : Actors and actresses : Academy awards (Motio
    - Headline News 01/08/98
    - Headline News 01/09/98
    - Headline News 01/10/98

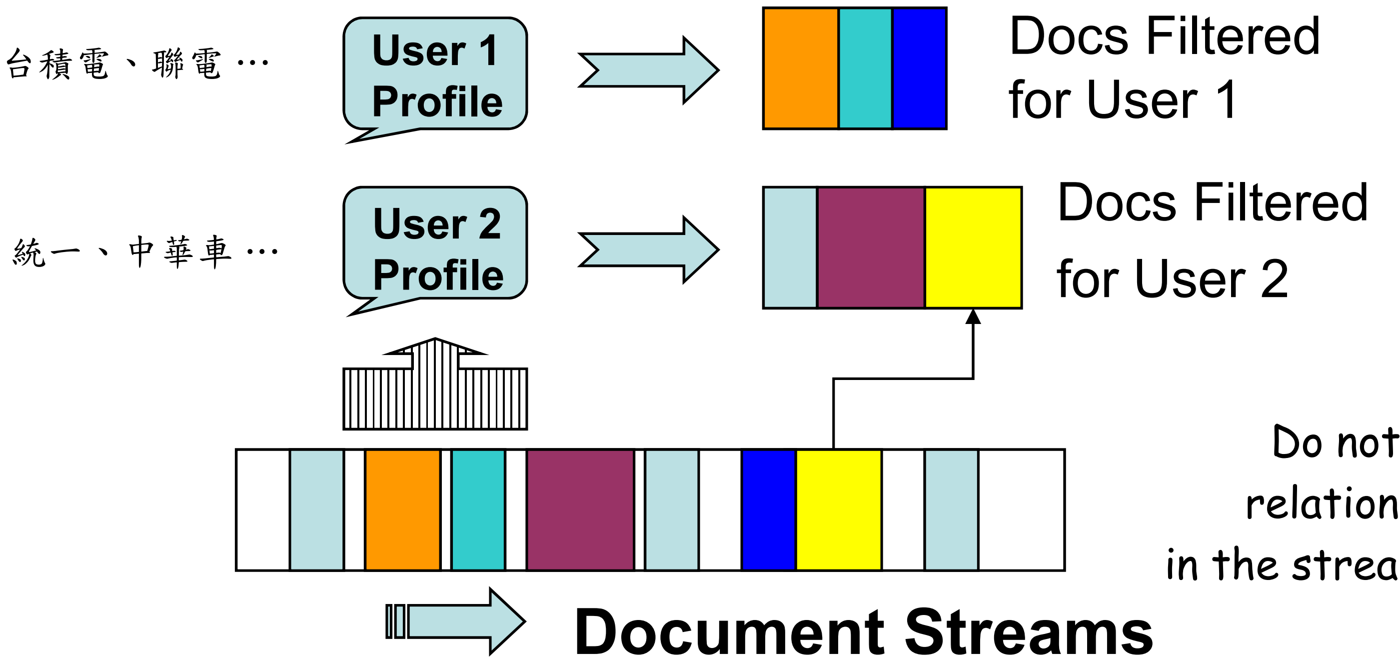
Retrieval: Ad Hoc

- *Ad hoc* retrieval
  - Documents remain relatively static while new queries are submitted to the system
    - The statistics for the entire document collection is obtainable
  - The most common form of user task



Retrieval: Filtering

- Filtering
  - – Queries remain relatively static while new documents come into the system (and leave)
    - User profiles: Describe the users' preferences
  - – E.g. news wiring services in the stock market



Do not consider the relations of documents in the streams (only **user task**)

Filtering & Routing

- **Filtering** task indicates to the user which document might be interested to him
    - Determine which ones are really relevant is fully reserved to the user
        - Documents with a ranking about a given threshold is selected
    - But no ranking information of filtered documents is presented to user
- **Routing**: a variation of filtering
    - Ranking information of the filtered documents is presented to the user
    - The user can examine the Top N documents
- The vector model is preferred (simplicity!)
    - For filtering/routing, the crucial step is not ranking but the construction of user profiles

Filtering: User Profile Construction

- Simplistic approach
  - Describe the profile through a set of keywords
  - The user provides the necessary keywords
  - User is not involved too much
  - Drawback: If user not familiar with the service (e.g. the vocabulary of upcoming documents)
- Elaborate approach
  - Collect information from user the about his preferences
  - Initial (primitive) profile description is adjusted by relevance feedback (from relevant/irrelevant information)
    - User intervention
  - Profile is continue changing

A Formal Characterization of IR Models

- The quadruple /**D**, **Q**, F, $R(q_i,d_j)$/ definition
  - **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, i.e., “queries”
  - F: a framework for modeling documents representations, queries, and their relationships and operations
  - $R(q_i, d_j)$: a ranking function which associations a real number with $q_i \in$ **Q** and $d_j \in$ **D**
    - Define an ordering among the documents d_j with regard to the query q_i

A Formal Characterization of IR Models (cont.)

- Classic Boolean model
  - Set of documents
  - Standard operations on sets

- Classic vector model
  - t-dimensional vector space
  - Standard linear algebra operations on vectors

- Classic probabilistic model
  - Sets (relevant/irrelevant document sets)
  - Standard probabilistic operations
    - Mainly the Bayes’ theorem

Classic IR Models - Basic Concepts

- Each document represented by a set of representative keywords or index terms
- An index term is a document word whose semantics is useful for remembering the document main themes
- Usually, index terms are nouns because nouns have meaning by themselves
  - Adjectives,adverbs, and connectives mainly work as complements
- However, search engines assume that all words are index terms (full text representation)

Classic IR Models - Basic Concepts (cont.)

- Not all terms are equally useful for representing the document contents
  - *less frequent terms* allow identifying a narrower set of documents
- The *importance* of the index terms is represented by *weights* associated to them
  - Let
    - k_i be an index term
    - d_j be a document
    - w_{ij} be a weight associated with (k_i, d_j)
    - $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$: an index term vector for the document d_j
    - $g_i(\vec{d_j}) = w_{i,j}$
  - The weight w_{ij} quantifies the importance of the index term for describing the document semantic contents

Classic IR Models - Basic Concepts (cont.)

- Correlation of index terms
  - E.g.: computer and network
  - Consideration of such correlation information does not consistently improve the final ranking result
    - Complex and slow operations

- Important Assumption/Simplification
  - Index term weights are mutually independent ! (bag-of-words modeling)
  - However, the appearance of one word often attracts the appearance of the other (e.g., “Computer” and “Network”)

The Boolean Model

- Simple model based on set theory and Boolean algebra
- A query specified as boolean expressions with and, or, not operations (connectives)
  - Precise semantics, neat formalism and simplicity
  - Terms are either present or absent, i.e., $w_{ij} \in \{0,1\}$
- A query can be expressed as a disjunctive normal form (DNF) composed of conjunctive components
  - $\vec{q}_{dnf}$: the DNF for a query q
  - $\vec{q}_{cc}$: conjunctive components (binary weighted vectors) of $\overrightarrow{q_{dnf}}$

The Boolean Model (cont.)

- For intance, a query [$q = k_a \wedge (k_b \vee \neg k_c)$] can be written as a DNF

$$\vec{q}_{dnf}=(1,1,1) \vee (1,1,0) \vee (1,0,0)$$

conjunctive components

(binary weighted vectors)

a canonical representation

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

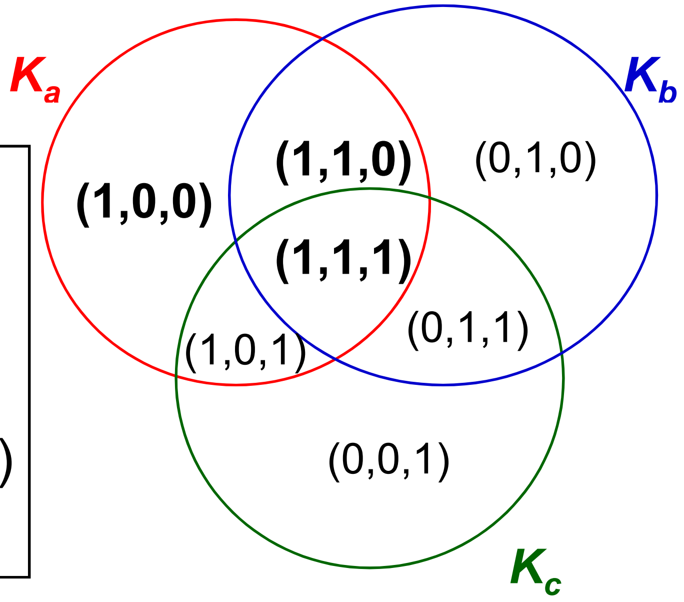
$=(k_a \wedge k_b) \vee (k_a \wedge \neg k_c)$

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

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

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

$=> \vec{q}_{dnf}=(1,1,1) \vee (1,1,0) \vee (1,0,0)$



The Boolean Model (cont.)

- The similarity of a document d_j to the query q (i.e., premise of relevance)

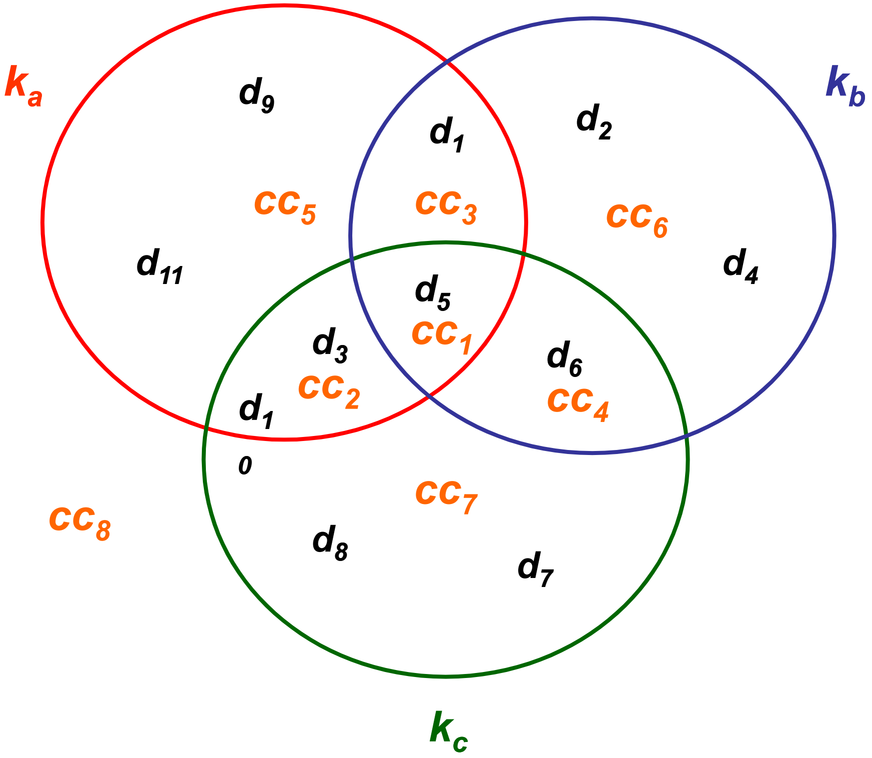
$$sim(d_j,q) = \begin{cases} 1: \text{if } \exists \overrightarrow{q_{cc}} \mid (\overrightarrow{q_{cc}} \in \overrightarrow{q_{dnf}} \wedge (\forall k_i, g_i(\overrightarrow{d_j}) = g_i(\overrightarrow{q_{cc}})) \\ 0: \text{otherwise} \end{cases}$$

A document is represented as
a conjunctive normal form

  - $sim(d_j,q)=1$ means that the document d_j is relevant to the query q
  - Each document d_j can be represented as a conjunctive component (vector)

Advantages of the Boolean Model

- Simple queries are easy to understand relatively easy to implement (simplicity and neat model formulation)
- The dominant language (model) in commercial (bibliographic) systems until the WWW



$cc_1 = k_a\ k_b\ k_c$
$cc_2 = k_a\ \overline{k_b}\ k_c$
$cc_3 = k_a\ k_b\ \overline{k_c}$
$cc_4 = \overline{k_a}\ k_b\ k_c$
$cc_5 = k_a\ \overline{k_b}\ \overline{k_c}$
$cc_6 = \overline{k_a}\ k_b\ \overline{k_c}$
$cc_7 = \overline{k_a}\ \overline{k_b}\ k_c$
$cc_8 = \overline{k_a}\ \overline{k_b}\ \overline{k_c}$

Drawbacks of the Boolean Model

- Retrieval based on binary decision criteria with no notion of partial matching (no term weighting)
  - No noton of a partial match to the query condition
  - No ranking (ordering) of the documents is provided (absence of a grading scale)
  - Term freqency counts in documents not considered
  - Much more like a data retrieval model

Drawbacks of the Boolean Model (cont.)

- Information need has to be translated into a Boolean expression which most users find awkward
  - The Boolean queries formulated by the users are most often too simplistic (difficult to specify what is wanted)
- As a consequence, the Boolean model frequently returns either too few or too many documents in response to a user query
- However, the Boolean model is still dominant model with commercial document database systems

The Vector Model

SMART system
Cornell U., 1968

- Also called *Vector Space Model (VSM)*

- Some perspectives
  - Use of *binary weights* is too limiting
  - *Non-binary weights* provide consideration for partial matches
  - These term weights are used to compute a *degree of similarity* between a query and each document
  - Ranked set of documents provides better matching for user information need

The Vector Model (cont.)

- Definition:
  - w_{ij} > =0 whenever $k_i \in d_j$
  - w_{iq} >= 0 whenever $k_i \in q$
  - document vector $\vec{d_j} = (w_{1j}, w_{2j}, ..., w_{tj})$
  - query vector $\vec{q} = (w_{1q}, w_{2q}, ..., w_{tq})$
  - To each term k_i is associated a unitary (basis) vector $\vec{u_i}$
  - The unitary vectors $\vec{u_i}$ and $\vec{u_s}$ are assumed to be **orthonormal** (i.e., index terms are assumed to occur independently within the documents)

totally t terms in the vocabulary

- The t unitary vectors $\vec{u_i}$ form an orthonormal basis for a t-dimensional space
  - Queries and documents are represented as weighted vectors

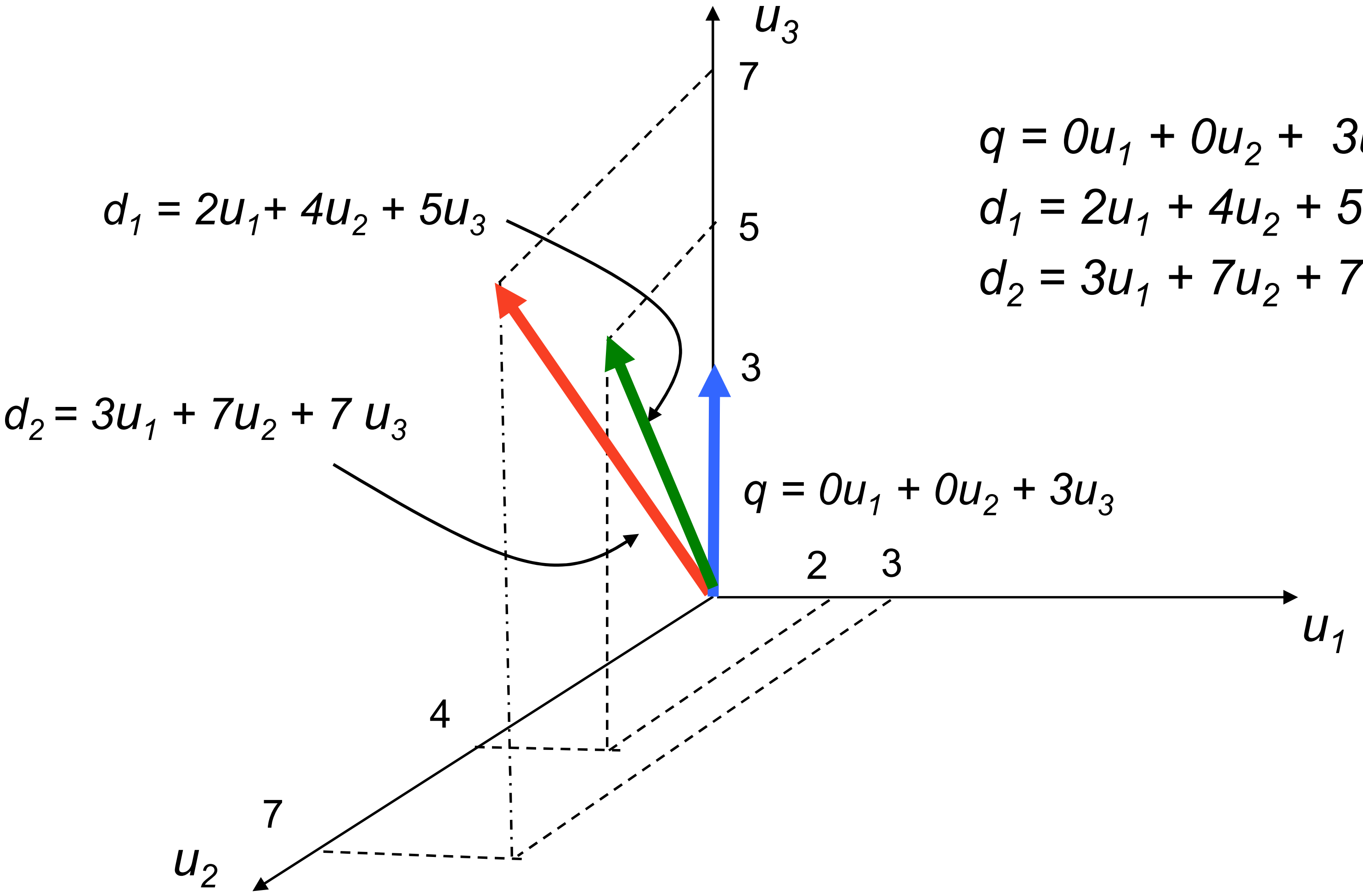
The Vector Model (cont.)

- How to measure the degree of similarity
  - Distance, angle or projection?

$q = 0u_1 + 0u_2 + \ 3u_3$

$d_1 = 2u_1 + 4u_2 + 5u_3$

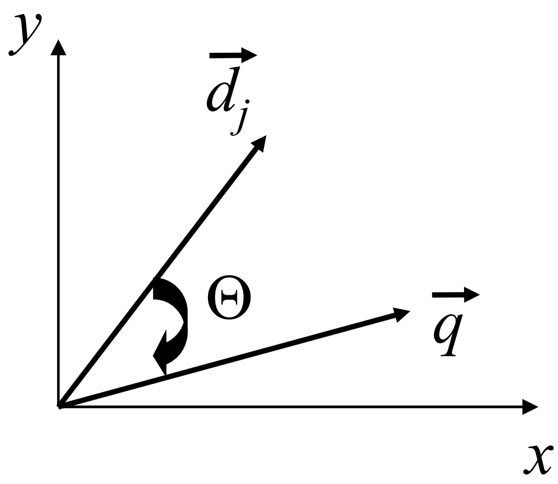
$d_2 = 3u_1 + 7u_2 + 7u_3$



The Vector Model (cont.)

- The similarity of a document d_j to the query q

$$
\begin{aligned}
& sim(d_j, q) \\
& = \text{cosine}(\Theta) \\
& = \frac{\vec{d}_j \bullet \vec{q}}{|\vec{d}_j| \times |\vec{q}|} \\
& = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}
\end{aligned}
$$



Document length normalization

The same for documents, can be discarded ⇨ Won’t affect the final ranking

(if discarded, equivalent to the projection of the query on the document vector)

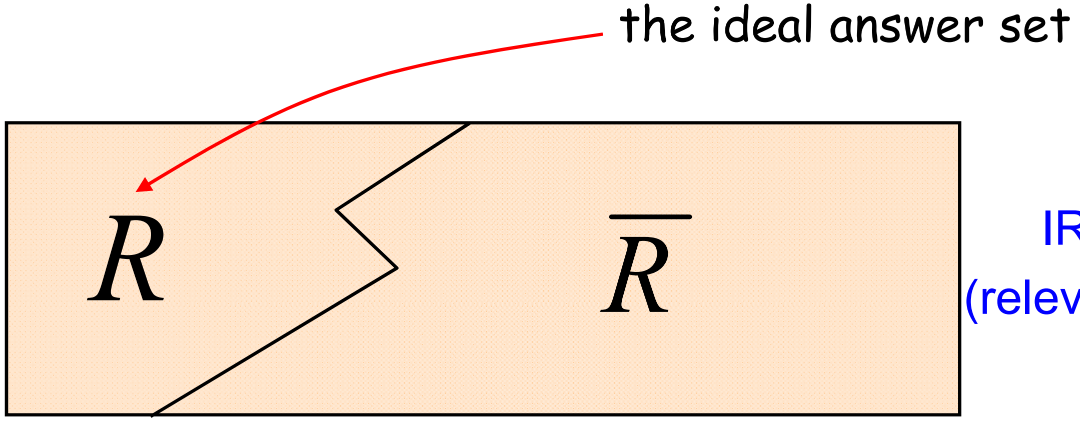
  - Establish a threshold on *sim*(d_j,*q*) and retrieve documents with a degree of similarity above the threshold

The Vector Model (cont.)

- Degree of similarity ⇨ Relevance
  - Usually, w_{ij} > =0 & w_{iq} >= 0
    - Cosine measure ranges between 0 and 1
  - $sim(d_j, q) \approx 1$ ⇨ highly relevant !
  - $sim(d_j, q) \approx 0$ ⇨ almost irrelevant !

The Vector Model (cont.)

- The role of index terms



IR as a binary clustering (relevant/non-relevant) problem

Document collection

- Which index terms (features) better describe the relevant class
  - Intra-cluster similarity (*tf*-factor)
  - Inter-cluster dissimilarity (*idf*-factor)

balance between these two factors

The Vector Model (cont.)

- How to compute the weights w_{ij} and w_{iq} ?
- A good weight must take into account two effects:
  - Quantification of **intra-document** contents (similarity)
    - *tf* factor, the **term frequency** within a document
    - High term frequency is needed
  - Quantification of **inter-documents** separation (dissimilarity)
    - Low **document frequency** is preferred
    - *idf* (*IDF*) factor, the **inverse document frequency**
  - $w_{i,j} = tf_{i,j} * idf_i$
  - =

Specifically, a term weighting mechanism should give a low weight to a high-frequent term that occurs in many documents and a high weight to a word that occurs in some documents but not all.

The Vector Model (cont.)

- Let,
  - N be the total number of docs in the collection
  - n_i be the number of docs which contain k_i
  - $freq_{i,j}$ raw frequency of k_i within d_j

- A normalized *tf* factor is given by

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

  - Where the maximum is computed over all terms which occur within the document d_j
  - $tf_{i,j}$ will be in the range of 0 to 1

The Vector Model (cont.)

- The *idf* factor is computed as

Sparck Jones

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

Document frequency of term k_i = $\frac{n_i}{N}$

  - The *log* is used to make the values of *tf* and *idf* comparable. It can also be interpreted as the amount of information associated with the term k_i

- The best term-weighting schemes use weights which are give by (for a term k_i in a document d_j)

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

  - The strategy is called a *tf-idf* weighting scheme

The Vector Model (cont.)

- For the query term weights, a suggestion is

Salton & Buckley

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

- The vector model with *tf-idf* weights is a good ranking strategy with *general* collections

- The vector model is usually as good as the known ranking alternatives. It is also simple and fast to compute

The Vector Model (cont.)

- Advantages
  - *Term-weighting* improves quality of the answer set
  - *Partial matching* allows retrieval of docs that approximate the query conditions
  - Cosine ranking formula sorts documents according to degree of similarity to the query

- Disadvantages
  - Assumes mutual independence of index terms
    - Not clear that this is bad though (??)

The Vector Model (cont.)

- Another *tf-idf* term weighting scheme
  - For query *q*

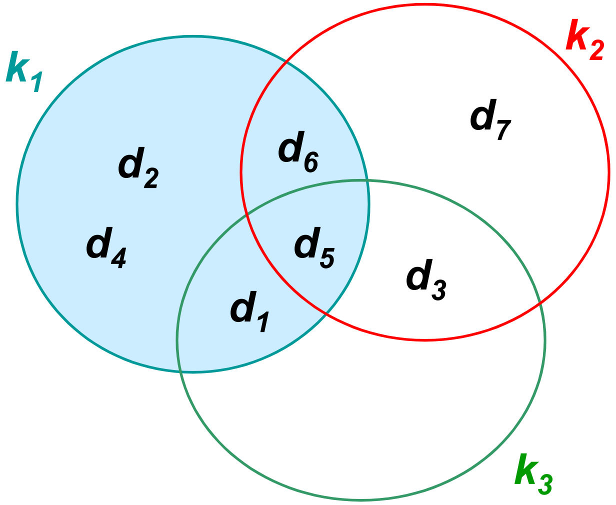
$$w_{i,q} = \underbrace{(1+\log(freq_{i,q}))}_{\text{Term Frequency}} \cdot \underbrace{\log((N+1)/n_i)}_{\text{Inverse Document Frequency}}$$

  - For document d_j

$$w_{i,j} = (1+\log(freq_{i,j}))$$

The Vector Model (cont.)

- Example



	k_1	k_2	k_3	$q \bullet d_j$	$q \bullet d_j/\|d\|$
d_1	1	0	1	**2**	**$2/\sqrt{2}$**
d_2	1	0	0	**1**	**$1/\sqrt{1}$**
d_3	0	1	1	**2**	**$2/\sqrt{2}$**
d_4	1	0	0	**1**	**$1/\sqrt{1}$**
d_5	1	1	1	**3**	**$3/\sqrt{3}$**
d_6	1	1	0	**2**	**$2/\sqrt{2}$**
d_7	0	1	0	**1**	**$1/\sqrt{1}$**
q	1	1	1		

The Vector Model (cont.)

- Experimental Results on TDT Chinese collections
  - Mandarin Chinese broadcast news
  - Measured in *mean* Average Precision (*m*AP)
  - ACM TALIP (2004)

Retrieval Results for the Vector Space Model

Average Precision		Word-level		Syllable-level	
		S(N), *N*=1	*S(N)*, *N*=1~2	*S(N)*, *N*=1	*S(N)*, *N*=1~2
TDT-2 (Dev.)	TD	0.5548	0.5623	0.3412	0.5254
	SD	0.5122	0.5225	0.3306	0.5077
TDT-3 (Eval.)	TD	0.6505	0.6531	0.3963	0.6502
	SD	0.6216	0.6233	0.3708	0.6353

$$R(q,d)=\sum_{j} w_j \cdot R_j(\vec{q}_j,\vec{d}_j),$$

types of index terms

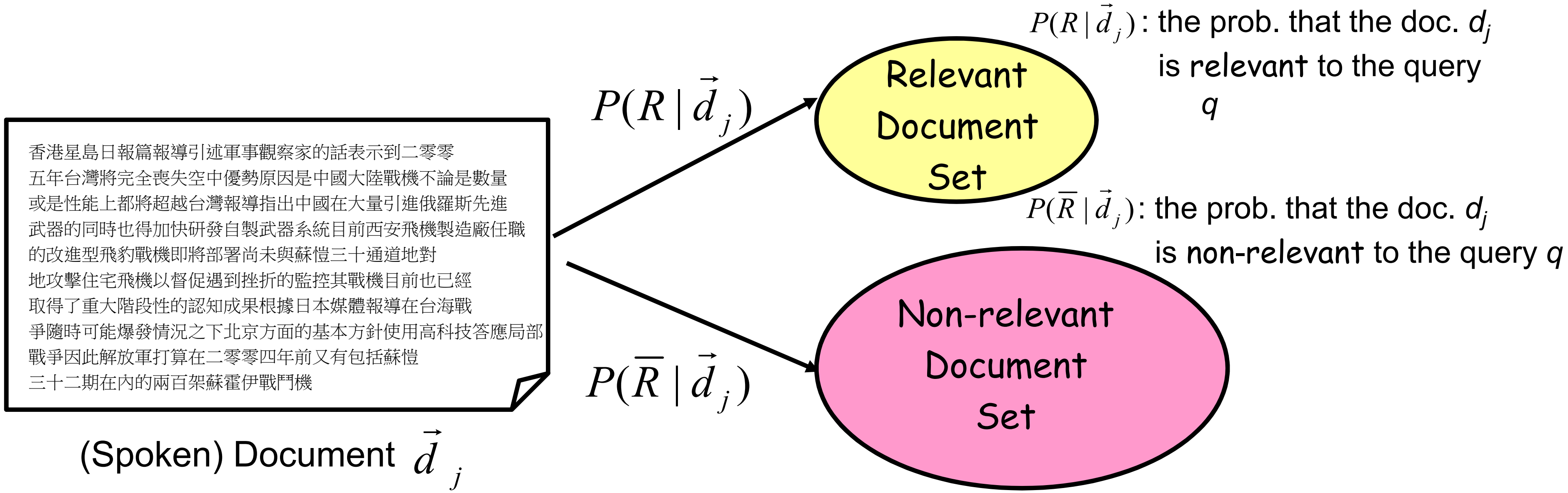
The Probabilistic Model

Roberston & Sparck Jones 1976

- Known as the Binary Independence Retrieval (BIR) model
  - "**Binary**": all weights of index terms are binary (0 or 1)
  - "**Independence**": index terms are independent !
- Capture the IR problem using a probabilistic framework
  - Bayes' decision rule

The Probabilistic Model (cont.)

- Retrieval is modeled as a *classification* process
  - Two classes for each query: the relevant or non-relevant documents



香港星島日報篇報導引述軍事觀察家的話表示到二零零
五年台灣將完全喪失空中優勢原因是中國大陸戰機不論是數量
或是性能上都將超越台灣報導指出中國在大量引進俄羅斯先進
武器的同時也得加快研發自製武器系統目前西安飛機製造廠任職
的改進型飛豹戰機即將部署尚未與蘇愷三十通道地對
地攻擊住宅飛機以督促遇到挫折的監控其戰機目前也已經
取得了重大階段性的認知成果根據日本媒體報導在台海戰
爭隨時可能爆發情況之下北京方面的基本方針使用高科技答應局部
戰爭因此解放軍打算在二零零四年前又有包括蘇愷
三十二期在內的兩百架蘇霍伊戰鬥機

(Spoken) Document $\vec{d}_j$

$P(R \mid \vec{d}_j)$: the prob. that the doc. d_j is relevant to the query q

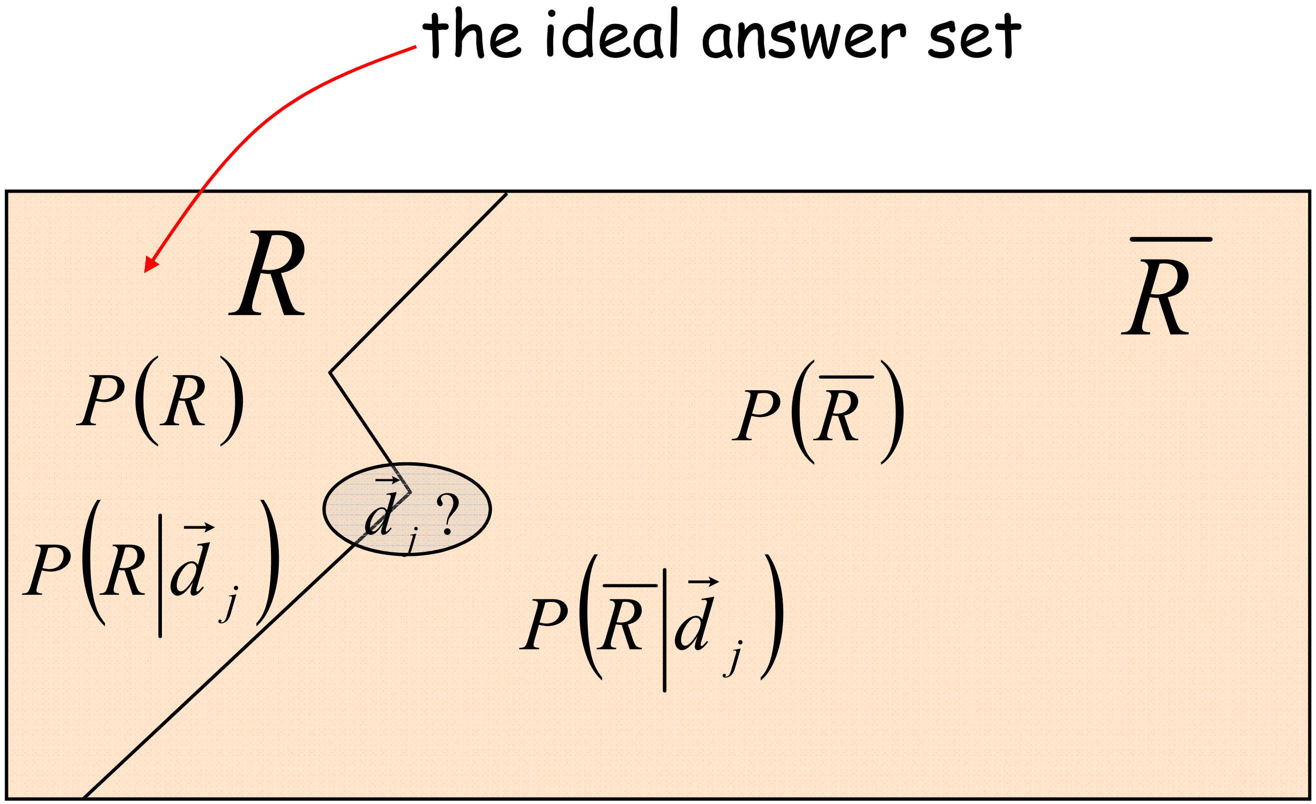
$P(\overline{R} \mid \vec{d}_j)$: the prob. that the doc. d_j is non-relevant to the query q

The Probabilistic Model (cont.)

- Given a user query, there is an *ideal answer set*
  - The querying process as a specification of the properties of this ideal answer set
- Problem: what are these properties?
  - Only the *semantics of index terms* can be used to characterize these properties
- **Guess at the beginning** what they could be
  - I.e., an initial guess for the preliminary probabilistis description of ideal answer set
- **Improve/Refine the probabilistic description of the answer set by iterations/interations**
  - Without (or with) the assistance from a human subject

The Probabilistic Model (cont.)

- How to improve the probabilistic description of the ideal answer set ?



Document Collection

The Probabilistic Model (cont.)

- Given a particular document d_j , calculate the probability of belonging to the relevant class, retrieve if greater than probability of belonging to non-relevant class

$$P(R \mid \vec{d}_j) > P(\bar{R} \mid \vec{d}_j)$$

Bayes' Decision Rule

- The similarity of a document d_j to the query q

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

Likelihood/Odds Ratio Test

Bayes' Theory

The same for all documents

$$= \frac{P(\vec{d}_j \mid R)P(R)}{P(\vec{d}_j \mid \bar{R})P(\bar{R})} \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \bar{R})}$$

$\geq \tau$?

if so, retrieved !

The Probabilistic Model (cont.)

- Explanation
  - $P(R)$: the prob. that a doc randomly selected form the entire collection is relevant
  - $P(\vec{d}_j \mid R)$: the prob. that the doc d_j is **relevant** to the query q (selected from the relevant doc set R)
- Further assume **independence** of index terms

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

$P(k_i \mid R)$: prob. that k_i is present in a doc randomly selected form the set R

$P(\overline{k}_i \mid R)$: prob. that k_i is not present in a doc randomly selected form the set R

$$P(k_i \mid R) + P(\overline{k}_i \mid R) = 1$$

$$\approx \frac{\left[\prod_{g_i(\vec{d}_j)=1} P(k_i \mid R)\right]\left[\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid R)\right]}{\left[\prod_{g_i(\vec{d}_j)=1} P(k_i \mid \overline{R})\right]\left[\prod_{g_i(\vec{d}_j)=0} P(\overline{k}_i \mid \overline{R})\right]}$$

The Probabilistic Model (cont.)

- Further assume *independence* of index terms
  - Another representation

$$sim\left(d_j, q\right) \approx \frac{\prod_{i=1}^{t}\left[P(k_i \mid R)^{g_i(\vec{d}_j)} P(\overline{k}_i \mid R)^{1-g_i(\vec{d}_j)}\right]}{\prod_{i=1}^{t}\left[P(k_i \mid \overline{R})^{g_i(\vec{d}_j)} P(\overline{k}_i \mid \overline{R})^{1-g_i(\vec{d}_j)}\right]}$$

  - Take logarithms

$$sim\left(d_j, q\right) \approx \log \frac{\prod_{i=1}^{t}\left[P(k_i \mid R)^{g_i(\vec{d}_j)} P(\overline{k}_i \mid R)^{1-g_i(\vec{d}_j)}\right]}{\prod_{i=1}^{t}\left[P(k_i \mid \overline{R})^{g_i(\vec{d}_j)} \left(P(\overline{k}_i \mid \overline{R})\right)^{1-g_i(\vec{d}_j)}\right]}$$

The same for all documents!

$$P(k_i \mid R) + P(\overline{k}_i \mid R) = 1$$
$$P(k_i \mid \overline{R}) + P(\overline{k}_i \mid \overline{R}) = 1$$

$$= \sum_{i=1}^{t} g_i\left(\vec{d}_j\right) \log \frac{P(k_i \mid R) P(\overline{k}_i \mid \overline{R})}{P(k_i \mid \overline{R}) P(\overline{k}_i \mid R)} + \cancel{\sum_{i=1}^{t} \log \frac{P(\overline{k}_i \mid R))}{P(\overline{k}_i \mid \overline{R})}}$$

$$= \sum_{i=1}^{t} g_i\left(\vec{d}_j\right)\left[\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})}\right]$$

The Probabilistic Model (cont.)

- Further assume **independence** of index terms
  - Use term weighting $w_{i,q}$ x $w_{i,j}$ to replace $g_i(\vec{d}_j)$

$$sim\left(d_j, q\right) \approx \sum_{i=1}^{t} g_i\left(\vec{d}_j\right)\left[\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})}\right]$$

$$\approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left[\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})}\right]$$

Binary weights (0 or 1) are used here

R is not known at the beginning

⇨ How to compute $P(k_i \mid R)$ and $P(k_i \mid \overline{R})$

The Probabilistic Model (cont.)

- Initial Assumptions
  - $P(k_i \mid R) = 0.5$:is constant for all indexing terms
  - $P(k_i \mid \overline{R}) = \frac{n_i}{N}$:approx. by distribution of index terms among all doc in the collection, i.e. the document frequency of indexing term k_i (Suppose that $|\overline{R}|>>|R|$, $N \approx |\overline{R}|$)

  (n_i: no. of doc that contain k_i. N : the total doc no.)
- Re-estimate the probability distributions
  - Use the initially retrieved and ranked Top *V* documents

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

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

V_i : the no. of documents in *V* that contain k_i

The Probabilistic Model (cont.)

- Handle the problem of “**zero**” probabilities
  - Add constants as the adjust constant

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

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

  - Or use the information of document frequency

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

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

The Probabilistic Model (cont.)

- Advantages
  - Documents are ranked in decreasing order of probability of relevance
- Disadvantages
  - Need to guess initial estimates for $P(k_i \mid R)$
  - Estimate the characteristics of the relevant class/set R through user-identified examples of relevant docs (without true training data)
  - All weights are binary: the method does not take into account *tf* and *idf* factors
  - Independence assumption of index terms

Brief Comparisons of Classic Models

- Boolean model does not provide for *partial matches* and is considered to be the weakest classic model

- Salton and Buckley did a series of experiments that indicated that, in general, the vector model outperforms the probabilistic model with *general collections*