



Lecture 4: Retrieval Models

01204453 Web Information Retrieval and Mining

Department of Computer Engineering
Faculty of Engineering, Kasetsart University
Bangkok, Thailand.



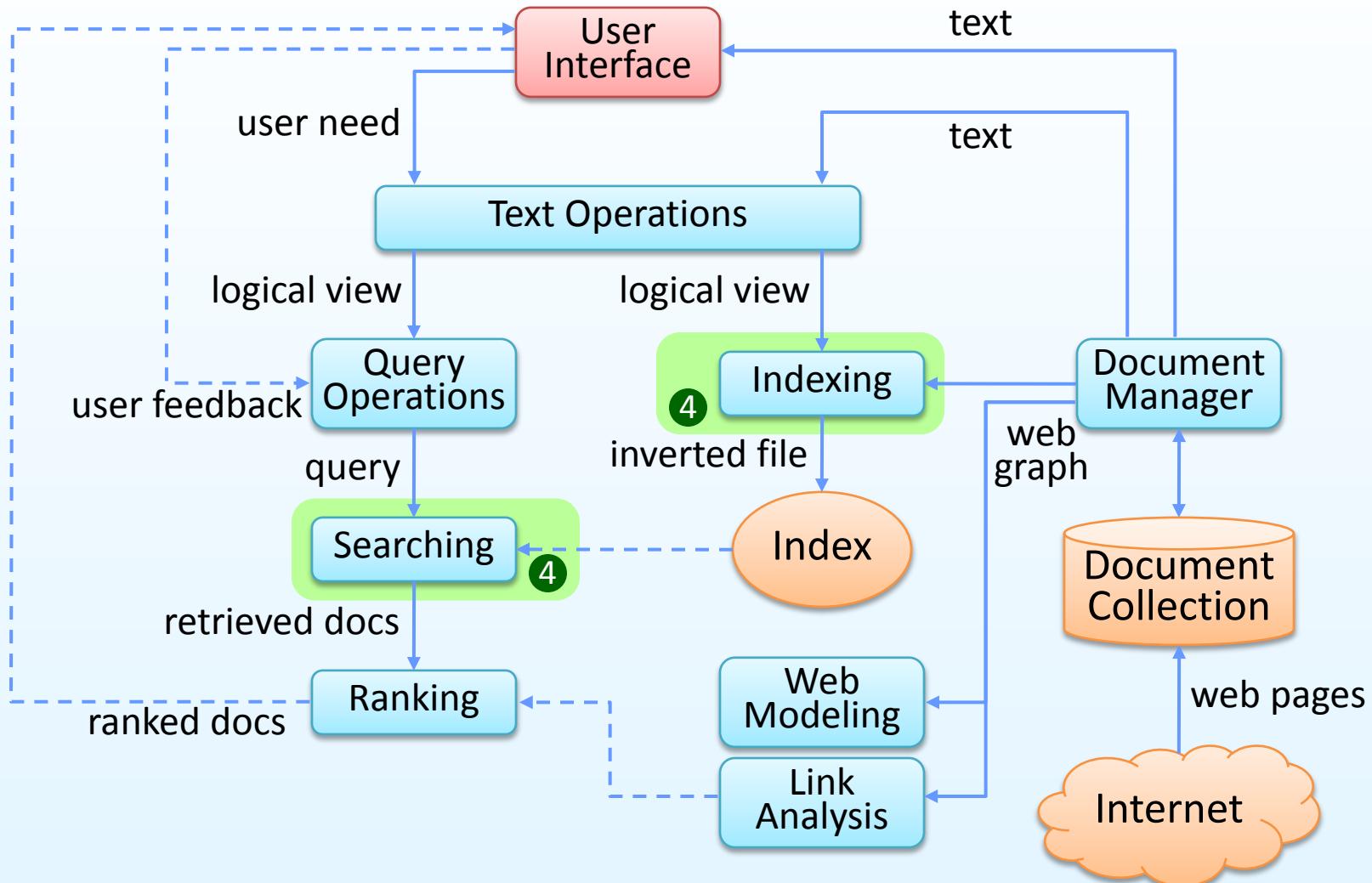
Department of
Computer Engineering
Kasetsart University



Outline

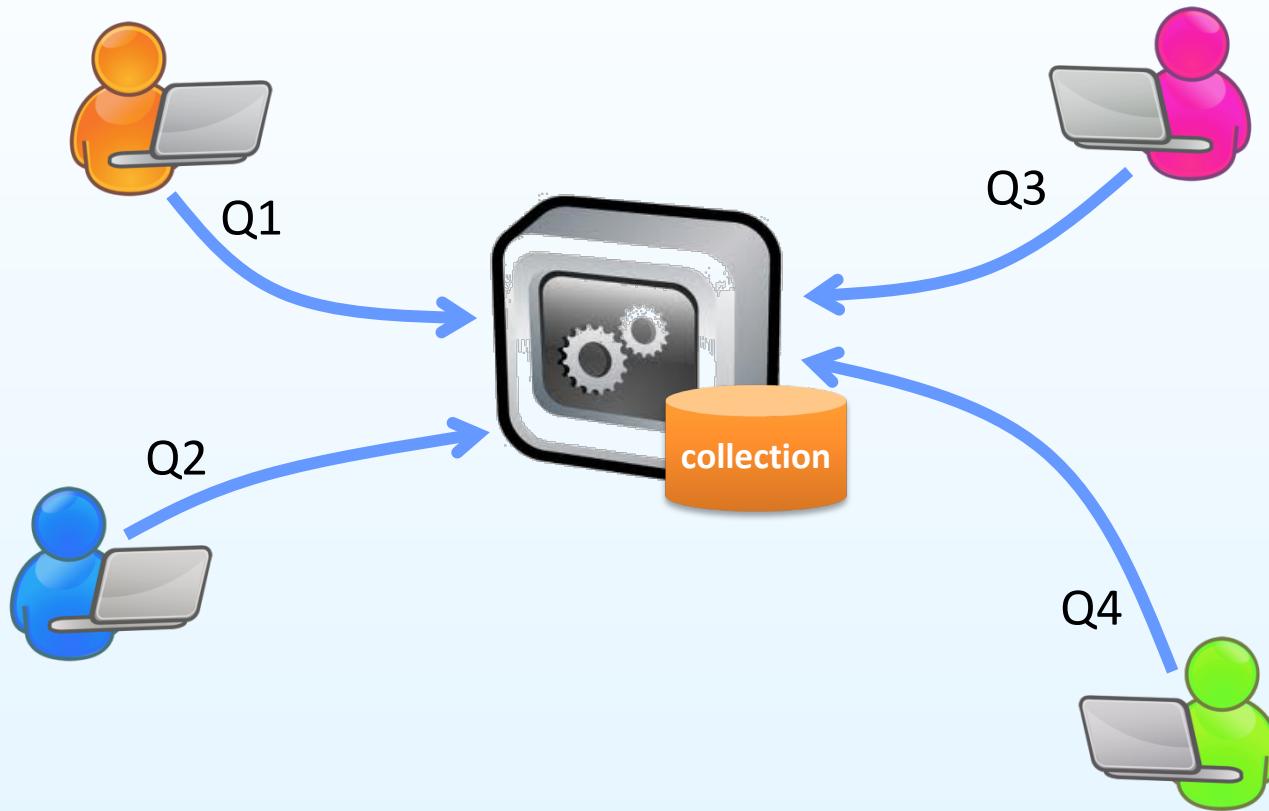
- Introduction to IR Models
- Basic Concepts
- Classical IR Models
 - Boolean Model
 - Vector Model
 - Probabilistic Model

Review: Search Engine Architecture



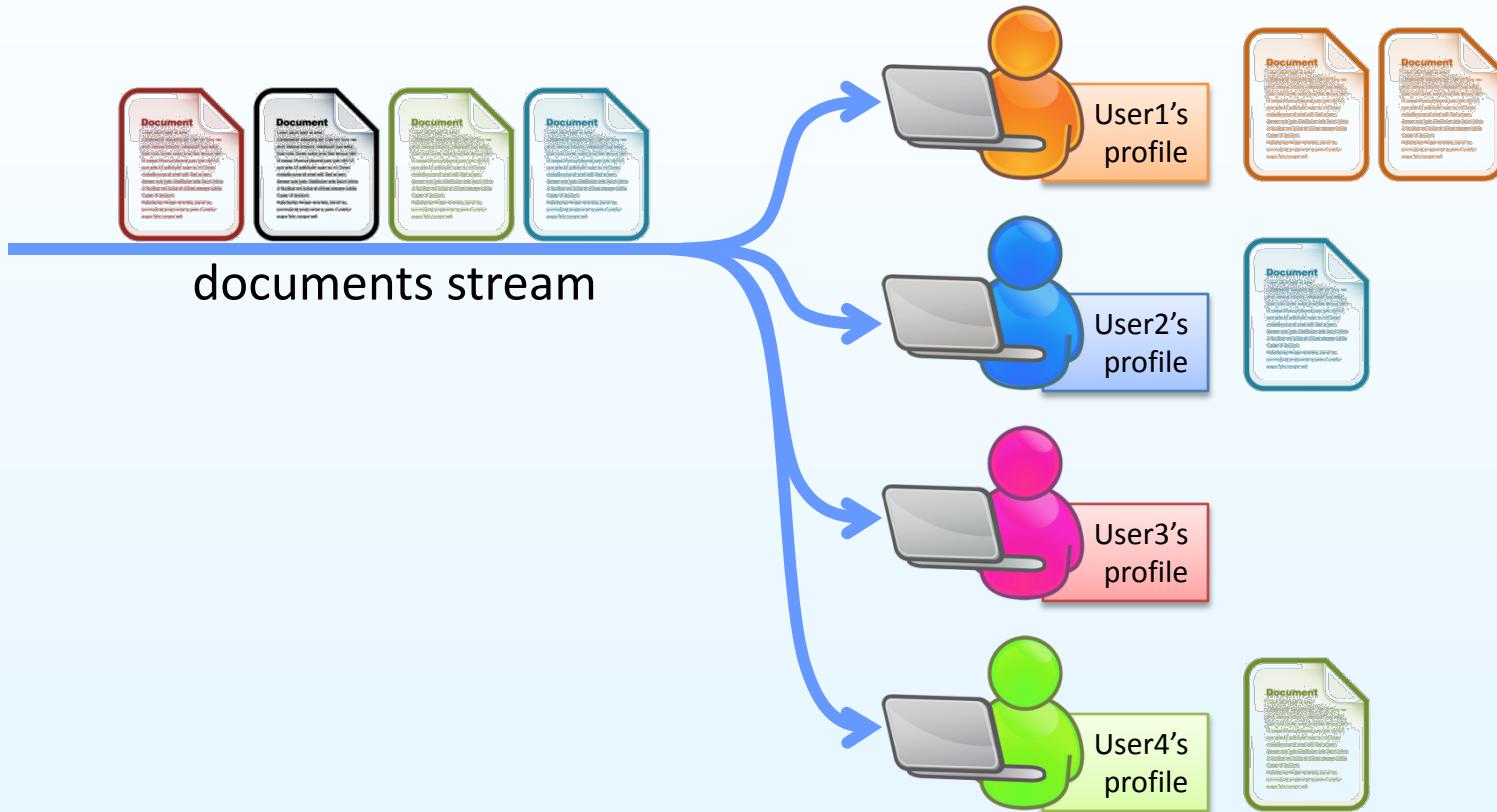
Retrieval: Ad Hoc VS. Filtering

- Ad hoc retrieval



Retrieval: Ad Hoc VS. Filtering

- Filtering



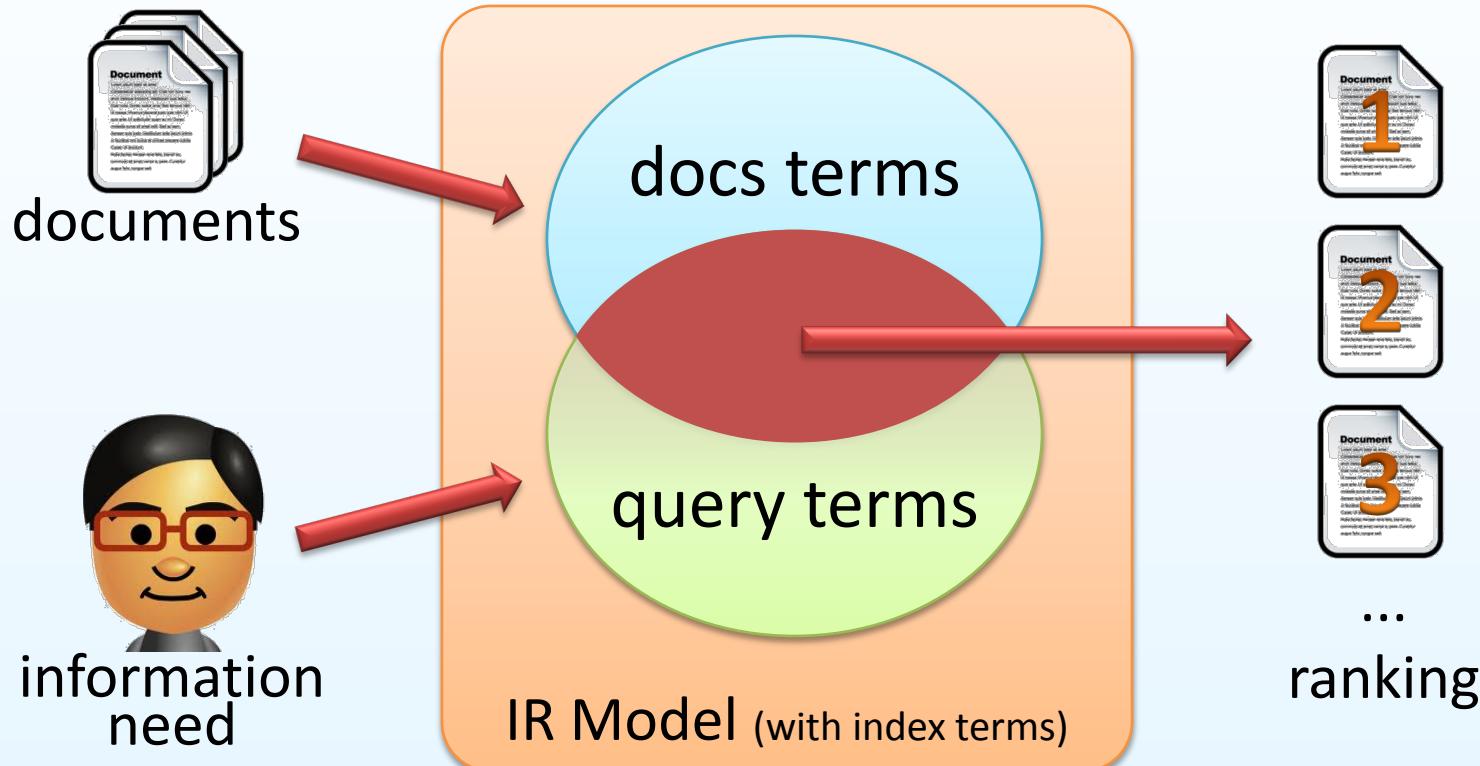
IR Models

- What is a model?
 - A representation of a system that allows for **investigation** of the **properties** of the system.
- 2 main components of modeling in IR:
 - A **logical framework** for representing documents and queries
 - A **ranking function** for quantifying similarities among documents and queries
- What is representative in indexing documents?
 - **Assumption:** the **semantics** of the documents and of the user information need can be naturally expressed through sets of **index terms**.

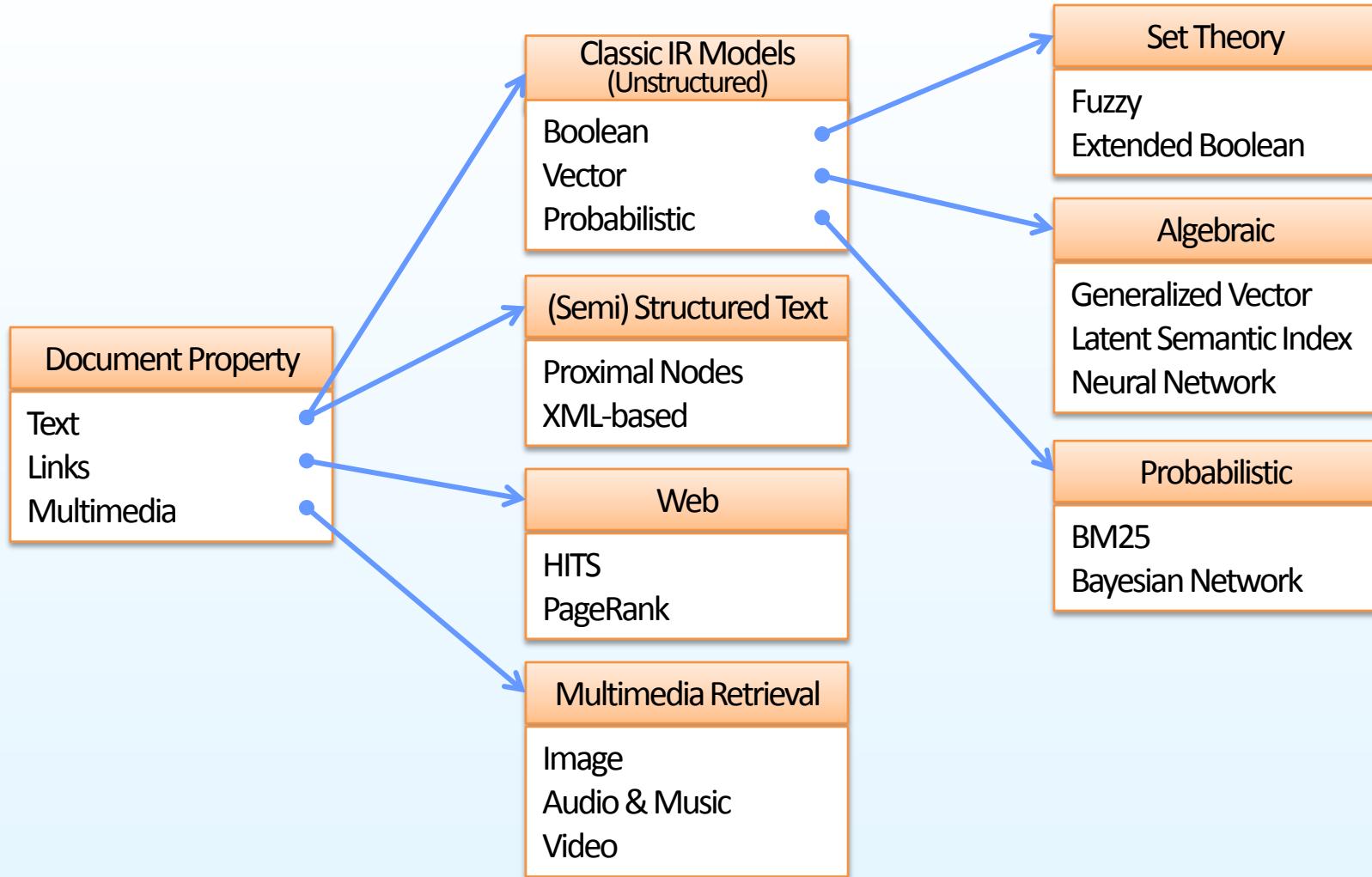
Index Terms

- Adopt to index and retrieve documents
- Usually play the role of noun—meaning on its own
- Appear in a document of a collection
- Also, simply refer to in a query

IR Process



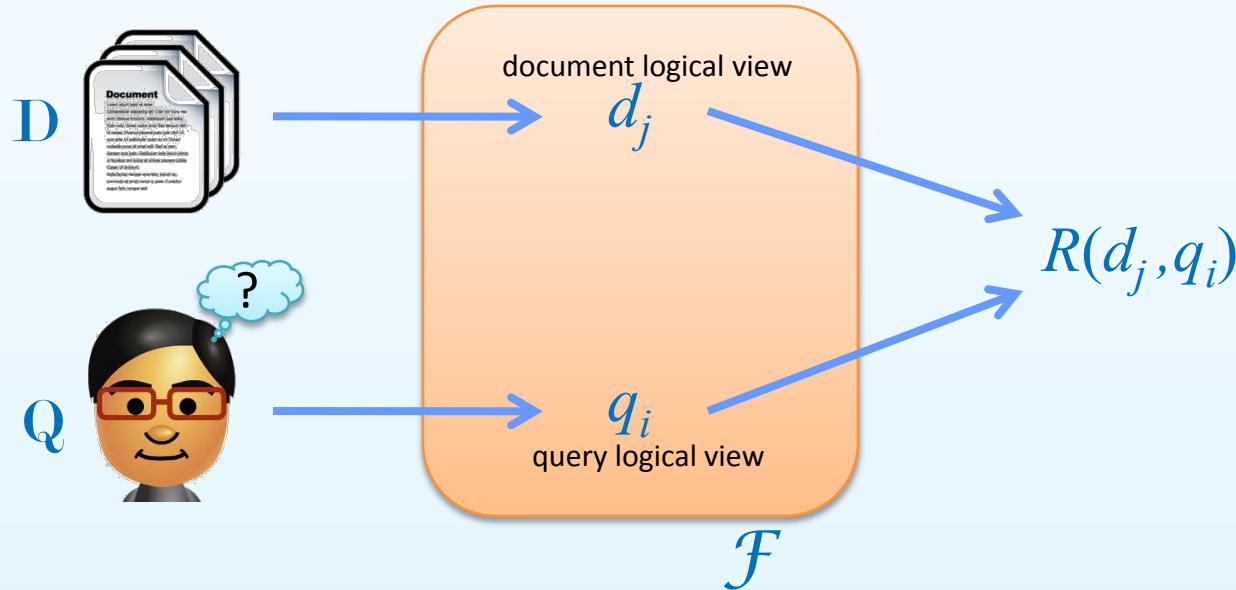
A Taxonomy of IR Models



Formal Characterization of IR Models

An **IR model** is a quadruple $[D, Q, \mathcal{F}, R(d_j, q_i)]$ where

- D is a set of logical views for the documents in the collection.
- Q is a set of logical views for the user queries.
- \mathcal{F} is a framework for modeling documents and queries.
- $R(d_j, q_i)$ is a ranking function.



Basic Concepts: Logical Views

- Each document is represented by a set of representative index terms (or keywords).
- An index term is a word or group of consecutive words in a document.
- It might assume that all word are index terms (full text representation).

Basic Concepts

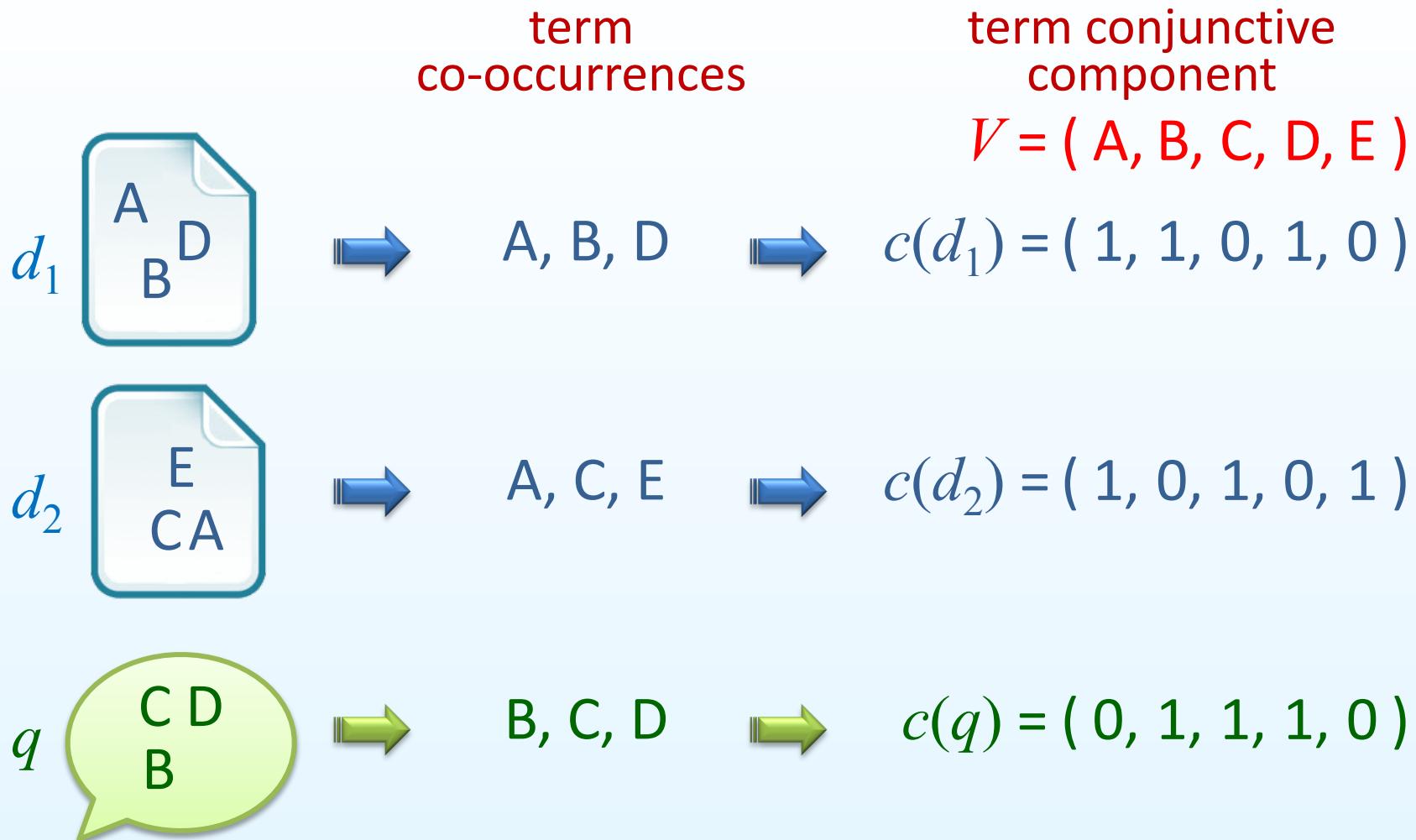
- Let
 - t be the number of distinct index terms in the document collection.
 - k_i be a generic index term.
- Then,
 - the vocabulary $V = \{k_1, k_2, \dots, k_t\}$ is the set of all distinct index terms in the collection.

Basic Concepts

- Documents and queries can be represented by **pattern of term co-occurrences**.
- Each of these patterns of term co-occurrence is called a **term conjunctive component**.
- For each document d_j (or query q), we associate a unique term conjunctive component $c(d_j)$ (or $c(q)$).
- Example:

$$V = \boxed{k_1 \ k_2 \ k_3 \ \dots \ k_t}$$
$$c(d_j) = \boxed{1 \ 1 \ 0 \ \dots \ 1}$$
$$c(q) = \boxed{0 \ 1 \ 0 \ \dots \ 0}$$

Term Conjunctive Component: Example



The Term-Document Matrix

- The occurrence of a term k_i in a document d_j establishes a relation between k_i and d_j .
- A **term-document relation** between k_i and d_j can be quantified by the frequency of them in the document.
- In a matrix form, this can written as:

$$\begin{matrix} & d_1 & d_2 & \dots & d_N \\ k_1 & freq_{1,1} & freq_{1,2} & \dots & freq_{1,N} \\ k_2 & freq_{2,1} & freq_{2,2} & \dots & freq_{2,N} \\ \dots & & & & \\ k_t & freq_{t,1} & freq_{t,2} & \dots & freq_{t,N} \end{matrix}$$

where each $freq_{i,j}$ element stands for the frequency of term k_i in document d_j .

Boolean Model

Boolean Model

- Simple model based on **set theory** and **boolean algebra**
- Queries specified as **boolean expressions**
 - Intuitive and precise semantics
 - Example:
$$q = k_a \wedge (k_b \vee \neg k_c)$$
- Frequencies in the term-document matrix are **all binary**.
 - $w_{i,j} \in \{0,1\}$: weight associated with pair (k_i, d_j)
 - $w_{i,q} \in \{0,1\}$: weight associated with pair (k_i, q)

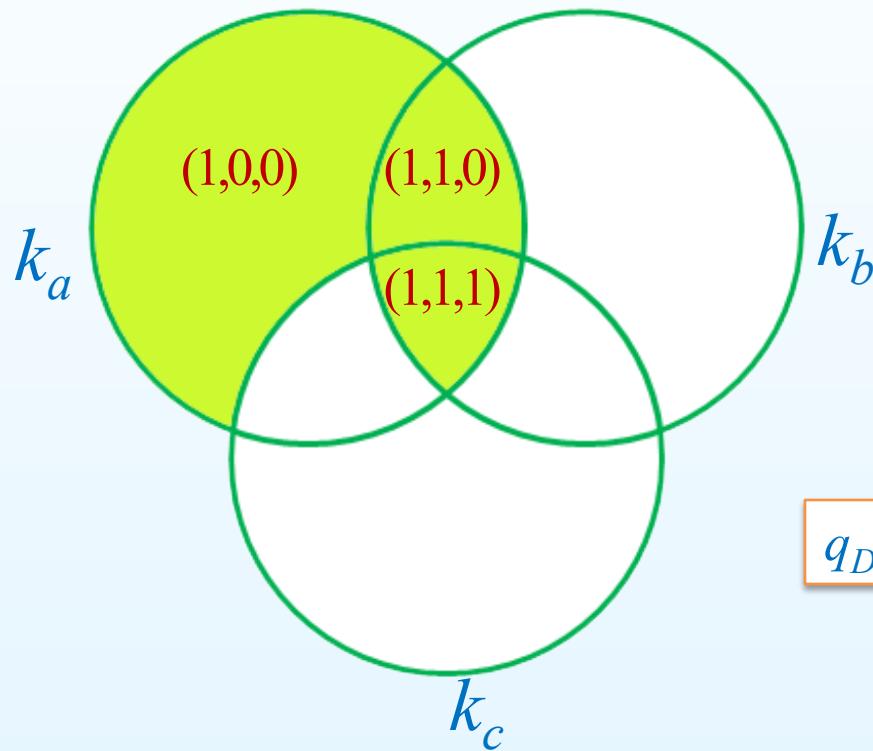
Boolean Model

- A term conjunctive component that satisfies a query q is called a **query conjunctive component** $c(q)$.
- A query q rewritten as a disjunction of those components is called the **disjunctive normal form** q_{DNF} .
- Let
 - the vocabulary $V = \{k_a, k_b, k_c\}$
 - a query $q = k_a \wedge (k_b \vee \neg k_c)$
- So that there are 3 query conjunctive components:
 - $(1,1,1)$, $(1,1,0)$, and $(1,0,0)$
- Then,
 - the disjunctive normal form
$$q_{DNF} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

Boolean Model

- The three conjunctive components for the query

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



$$q_{DNF} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

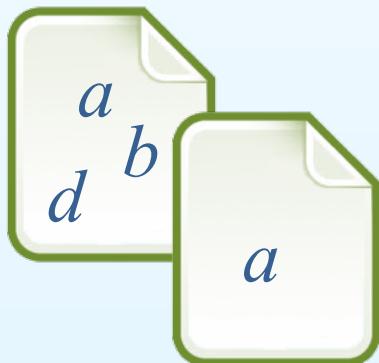
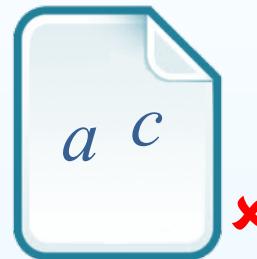
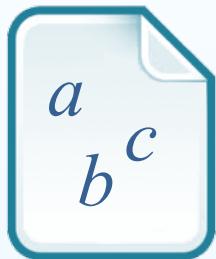
Boolean Model: Examples

- Which ones are the answers for $q = k_a \wedge (k_b \vee \neg k_c)$?

$$V = \{k_a, k_b, k_c\}$$



$$q_{DNF} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$



$$V = \{k_a, k_b, k_c, \color{red}{k_d}\}$$



$$q_{DNF} = ?$$

Boolean Model

- The similarity of the document d_j to the query q is defined as

$$sim(d_j, q) = \begin{cases} 1 & \text{if } \exists c(q) \in q_{DNF} \mid c(d_j) = c(q) \\ 0 & \text{otherwise} \end{cases}$$

- The boolean model predicts that each document is either **relevant** or **non-relevant (irrelevant)**.

Drawbacks of the Boolean Model

- Information need has to be translated into a boolean expression.
- Retrieval is based on binary decision criteria with **no partial matching**.
- **No ranking** (grading scale) of the documents is provided.
- The model frequently return either too few or too many documents in response to a user query.

Vector Model

Vector Model

- Boolean matching and binary weights are too limiting.
- The vector model proposes a framework in which **partial matching** is possible.
- This is accomplished by assigning **non-binary weights** to index terms in queries and in documents.
- Term weights are used to compute a **degree of similarity** between a query and each document.
- The documents are ranked in decreasing order of their degree of similarity.

Vector Model

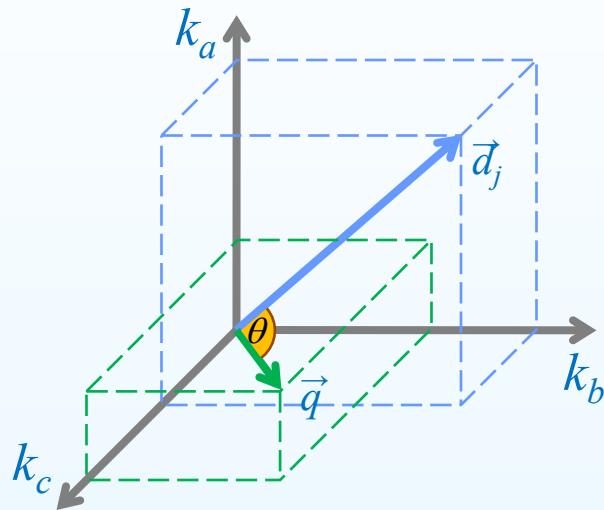
- For the vector model,
 - The **weight** $w_{i,j} > 0$ is associated to quantify the importance of the index term k_i in the documents d_j .
 - if k_i does not appear in d_j , then $w_{i,j} = 0$.
 - The index terms are assumed to be all **mutually independent**.
 - They are represented as unit vectors of a t -dimensional space, where t is the total number of index terms.
 - The representations of document d_j and query q are given by

$$\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$$

$$\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{t,q})$$

Vector Model

- Similarity between a document d_j and a query q

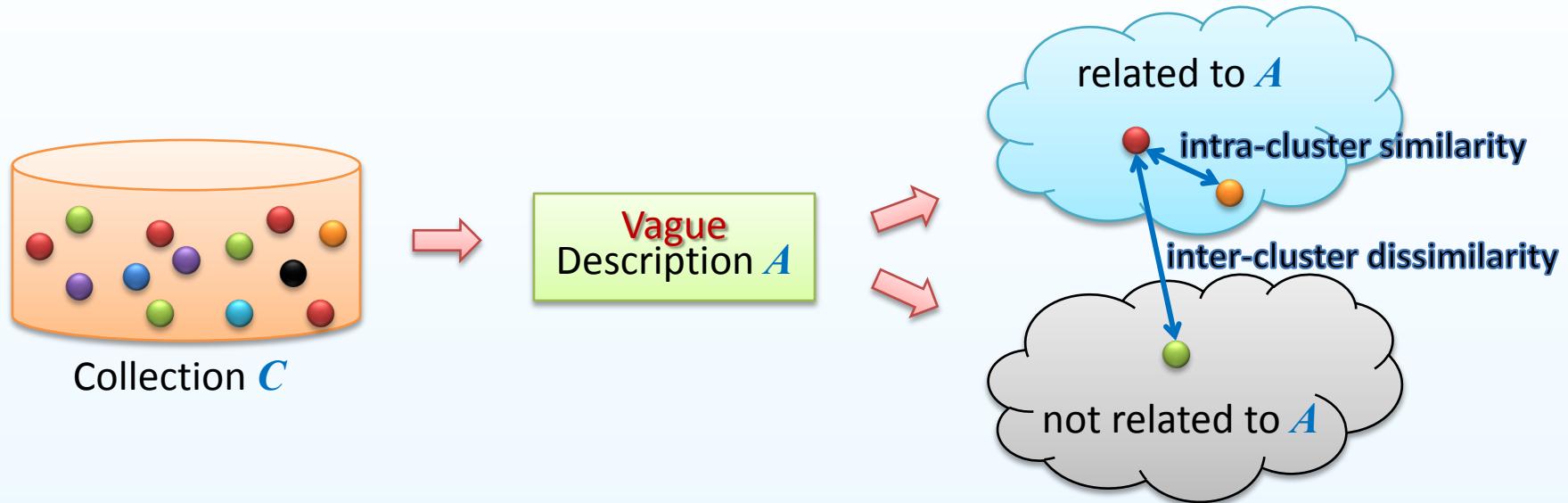


$$\cos(\theta) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \times |\vec{q}|}$$

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

Since $w_{i,j} \geq 0$ and $w_{i,q} \geq 0$, we have $0 \leq sim(d_j, q) \leq 1$.

IR Problem as a Clustering



- For the IR,
 - intra-cluster similarity is quantified by term frequency (tf)
 - inter-cluster dissimilarity is quantified by inverse document frequency (idf)

The *tf-idf* Weight

- Let
 - N be the total number of documents in the collection.
 - n_i be the number of documents in which the term k_i appears.
 - $\text{freq}_{i,j}$ be the raw frequency of term k_i in the document d_j .
- Then,
 - the normalized frequency of k_i in d_j is given by

$$tf_{i,j} = \frac{\text{freq}_{i,j}}{\max_l \text{freq}_{l,j}}$$

- the inverse document for k_i is given by

$$idf_i = \log_2 \frac{N}{n_i}$$

The *tf-idf* Weight

- The best known term-weighting schemes are

$$w_{i,j} = tf_{i,j} \times idf_i = \frac{freq_{i,j}}{\max_l freq_{l,j}} \times \log_2 \frac{N}{n_i}$$

- The query term weights

$$w_{i,q} = \left(0.5 + \frac{0.5 freq_{i,q}}{\max_l freq_{l,q}} \right) \times \log_2 \frac{N}{n_i}$$

Example: The *tf* Weights

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

d_1

To do is to be.
To be is to do.

d_2

To be or not to be.
I am what I am.

d_3

I think therefore I am.
Do be do be do.

d_4

Do do do, da da da.
Let it be, let it be.

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	1	1	-	-
2	do	0.5	-	1	1
3	is	0.5	-	-	-
4	be	0.5	1	0.667	0.667
5	or	-	0.5	-	-
6	not	-	0.5	-	-
7	I	-	1	0.667	-
8	am	-	1	0.333	-
9	what	-	0.5	-	-
10	think	-	-	0.333	-
11	therefore	-	-	0.333	-
12	da	-	-	-	1
13	let	-	-	-	0.667
14	it	-	-	-	0.667

Example: The *idf* Weights

$$idf_i = \log_2 \frac{N}{n_i}$$

d_1

To do is to be.
To be is to do.

d_2

To be or not to be.
I am what I am.

d_3

I think therefore I am.
Do be do be do.

d_4

Do do do, da da da.
Let it be, let it be.

Vocabulary		n_i	idf_i
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

Example: The *tf-idf* Weights

$$w_{i,j} = tf_{i,j} \times idf_i$$

d_1

To do is to be.
To be is to do.

d_2

To be or not to be.
I am what I am.

d_3

I think therefore I am.
Do be do be do.

d_4

Do do do, da da da.
Let it be, let it be.

Vocabulary		$w_{i,1}$	$w_{i,2}$	$w_{i,3}$	$w_{i,4}$
1	to	1	1	-	-
2	do	0.208	-	0.415	0.415
3	is	1	-	-	-
4	be	-	-	-	-
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	1	0.667	-
8	am	-	1	0.333	-
9	what	-	1	-	-
10	think	-	-	0.667	-
11	therefore	-	-	0.667	-
12	da	-	-	-	2
13	let	-	-	-	1.333
14	it	-	-	-	1.333

Example: The Vector Model

d_1

To do is to be.
To be is to do.

d_2

To be or not to be.
I am what I am.

d_3

I think therefore I am.
Do be do be do.

d_4

Do do do, da da da.
Let it be, let it be.

Consider a query “to do”:

$$q = (1, 0.415, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Doc.	Computation	Similarity
d_1	$\frac{(1 \times 1) + (0.208 \times 0.415)}{\sqrt{1 + 0.208^2} + 1 \times \sqrt{1 + 0.415^2}}$	0.702
d_2	$\frac{(1 \times 1) + (0 \times 0.415)}{\sqrt{6} \times \sqrt{1 + 0.415^2}}$	0.377
d_3	$\frac{(0 \times 1) + (0.415 \times 0.415)}{\sqrt{0.415^2 + 3 \cdot 0.667^2 + 0.333^2} \times \sqrt{1 + 0.415^2}}$	0.125
d_4	$\frac{(0 \times 1) + (0.415 \times 0.415)}{\sqrt{0.415^2 + 2^2 + 2 \cdot 1.333^2} \times \sqrt{1 + 0.415^2}}$	0.057

Document & Query Weighting Schemes

- SMART notation for $tf\text{-}idf$ variants

Term Frequency		Inverse Document Frequency		Normalization	
n (natural)	$freq_{i,j}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log_2(freq_{i,j})$	t (idf)	$\log_2\left(\frac{N}{n_i}\right)$	c (cosine)	$\frac{1}{\sqrt{\sum_{i=1}^t w_i^2}}$
a (augmented)	$0.5 + \frac{0.5 \times freq_{i,j}}{\max_l freq_{l,j}}$	p (prob. idf)	$\max\left(0, \log_2\left(\frac{N - n_i}{n_i}\right)\right)$	b (byte size)	$\frac{1}{CharLength^\alpha}, \alpha < 1$
b (boolean)	$\begin{cases} 1 & \text{if } freq_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$				
L (log avg.)	$\frac{1 + \log_2(freq_{i,j})}{1 + \log_2(\text{avg}(freq_{l,j}))}$				

Advantage & Disadvantage

- Advantages
 - Term-weighting improves quality of the answer set.
 - Partial matching allows retrieval of documents that approximate the query conditions.
 - Cosine ranking formula sorts documents according to a degree of similarity to the query.
 - Document length normalization is naturally built-in into the ranking.
- Disadvantage
 - The model assumes independence of index terms.

Any Question?