



A T M E
College of Engineering



Information Retrieval

Module 2

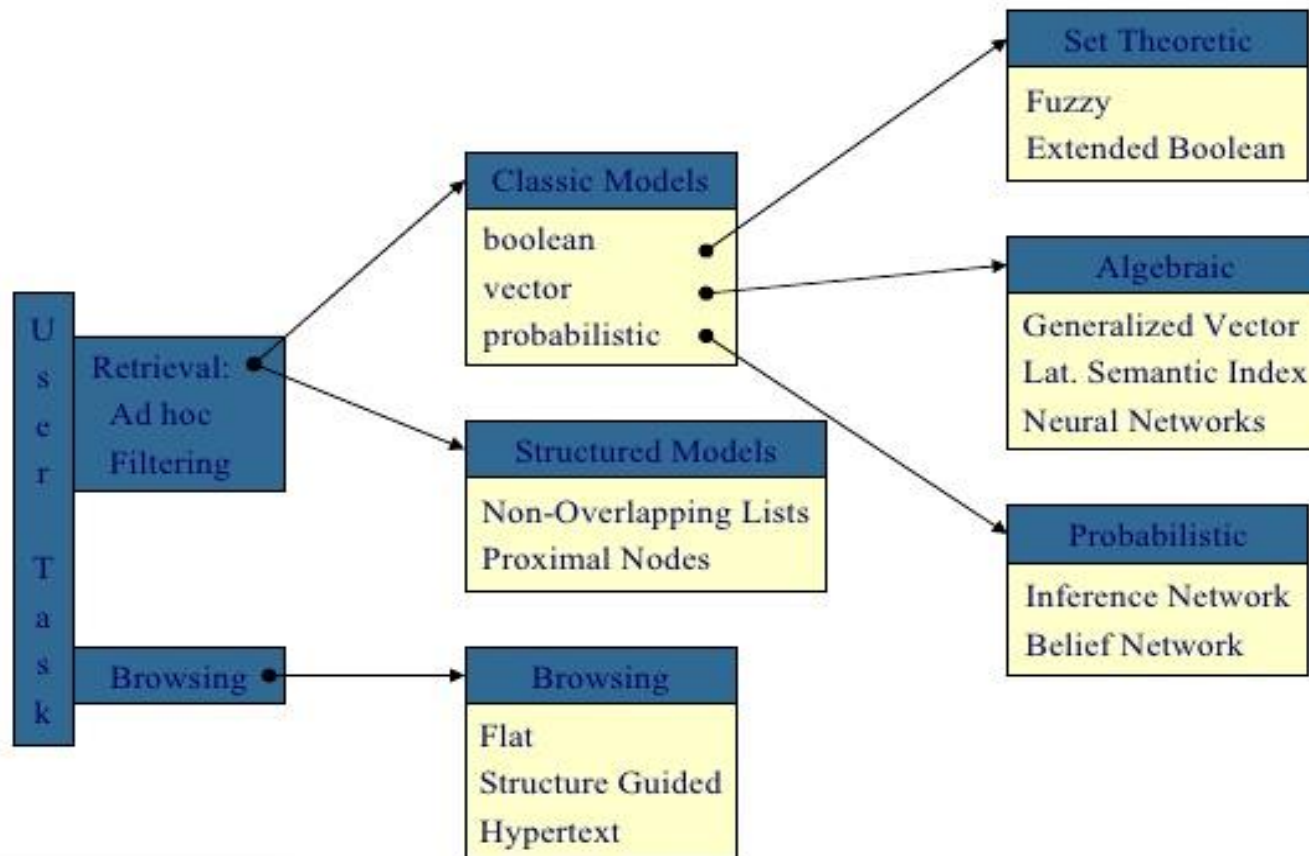
Modeling: A Taxonomy of IR models

- Three models:
 - Classic models
 - Structured models
 - Browsing models
- Three **classic models** in IR are :
 - Boolean: documents and queries are represented as sets of index terms. Also known as set theoretic.
 - Vector: documents and queries are represented as vectors in a t-dimensional space. Also known as algebraic.
 - Probabilistic: the framework for modeling document and query representations is based on probability theory.
- Set theoretic
 - Fuzzy
 - Extended boolean



- Algebraic
 - Generalized vector
 - Latent semantic indexing
 - Neural networks
- Probabilistic
 - Inference network
 - Belief network
- Two structured models in IR are
 - Non-overlapping lists model
 - Proximal nodes
- Three models for browsing are:
 - Flat
 - Structure guided
 - Hypertext

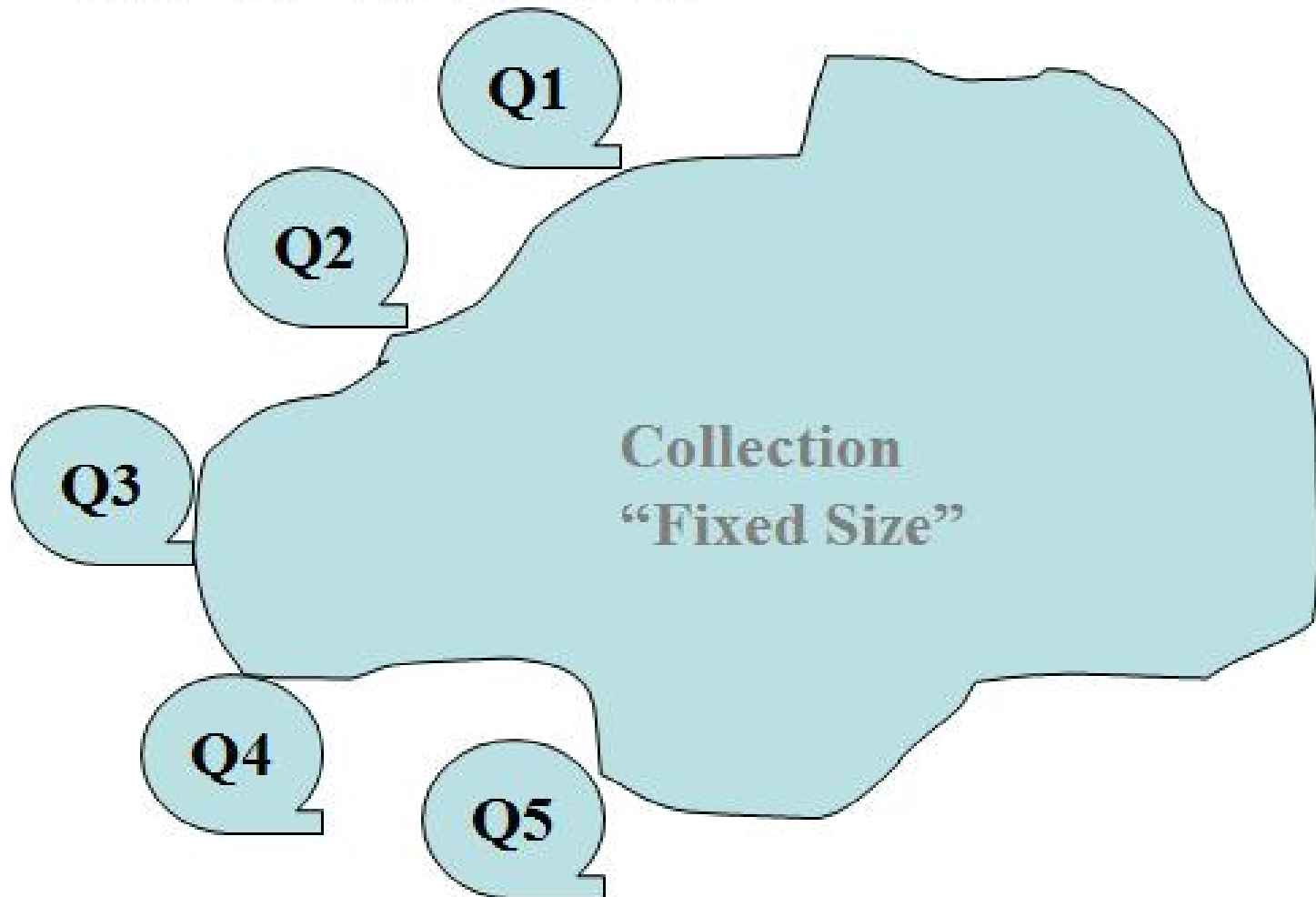
2.2 A Taxonomy of IR Models



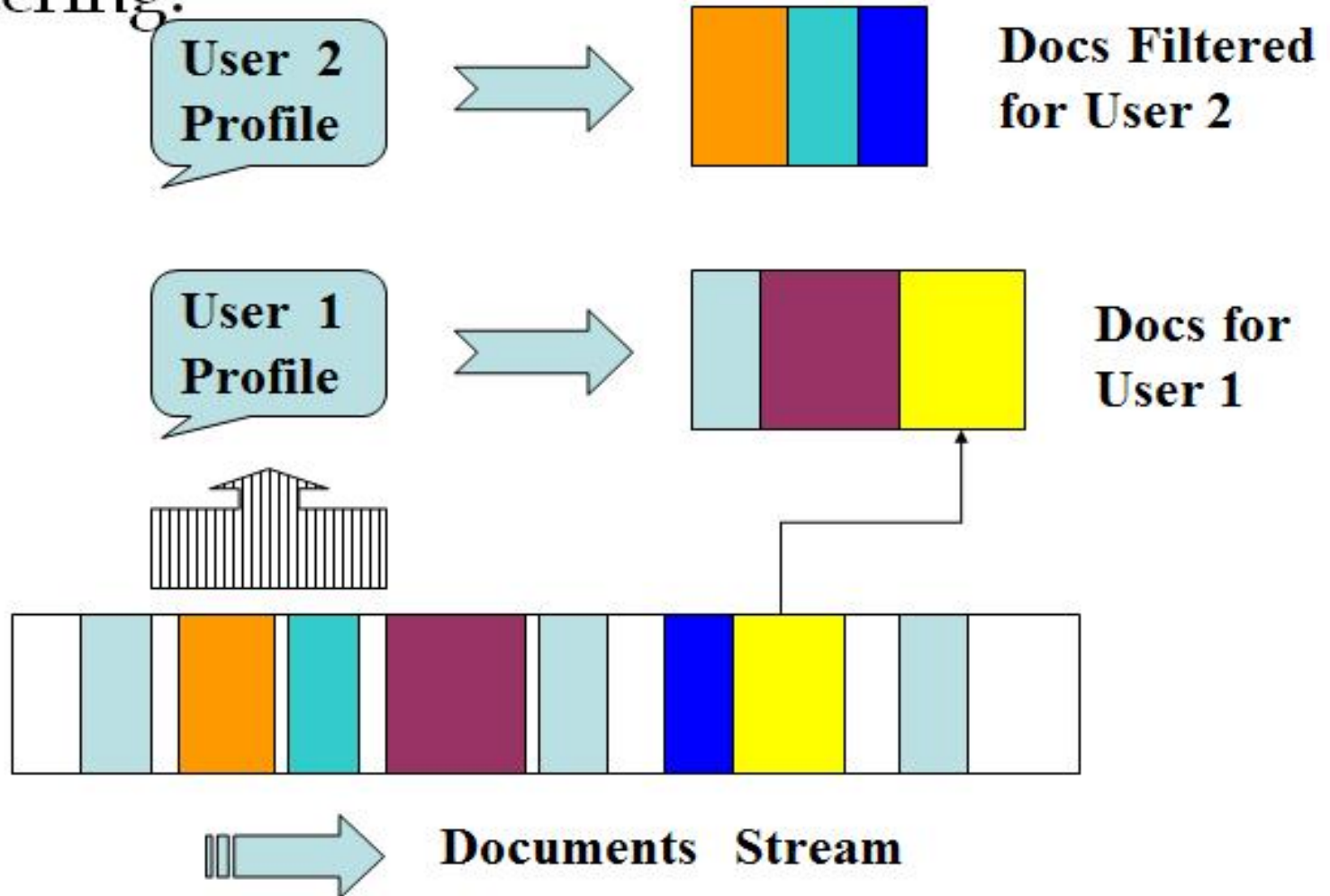


- Ad hoc retrieval
 - Standard retrieval task in which the user specifies his information need through a query which initiates a search (executed by the information system) for documents which are likely to be relevant to the user.
 - The documents in the collection remain relatively static while new queries are submitted to the system
 - The most common form of user task
- Filtering
 - The queries remain relatively static while new documents come into the system (and leave)
 - User profile
 - Describing the user's preferences
 - Routing (variation of filtering, rank the filtered document)

- Ad hoc retrieval:



- Filtering:





- Eg: selecting a news article among thousands of articles which are broadcast each day
- The task of determining which ones are really relevant is fully reserved to the user.
- This is accomplished by ranking the filtered documents and thereby user should examine smaller number of documents. This variation of filtering is called routing.
- To rank the documents vector model is preferred.
- In filtering, crucial task is not ranking, but the construction of the user profile.
- An approach for constructing a user profile is to describe the profile through a set of keywords and to require the user to provide the necessary keywords.

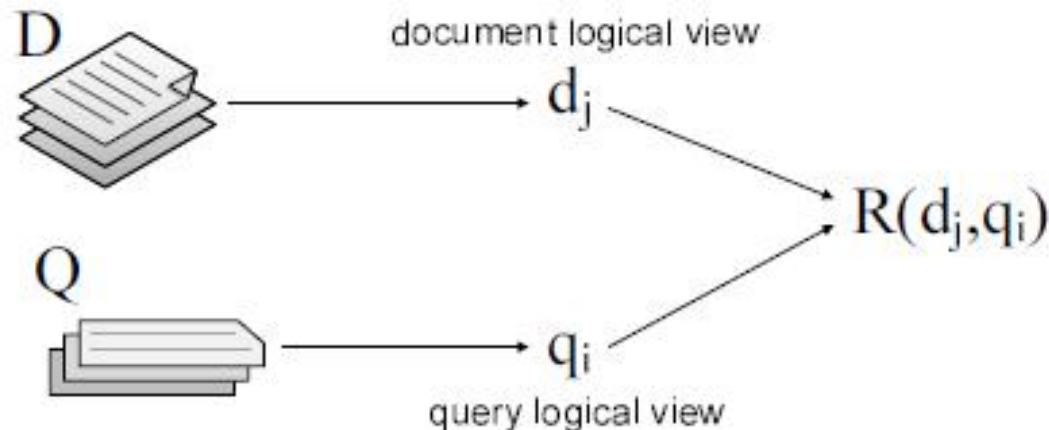


preferences and use this information to build the user profile.

- In the beginning user provides a set of keywords which describes an initial profile of his preferences.
- As new documents arrive the system uses this profile to select documents that are relevant and not relevant.
- The system uses this information to adjust the user profile description such that it reflects new preferences.

Formal characterization of IR models

- An **IR model** is a quadruple $[D, Q, F, R(q_i, d_j)]$ where
 1. D is a set of logical views for the documents in the collection
 2. Q is a set of logical views for the user queries
 3. F is a framework for modeling documents and queries
 4. $R(q_i, d_j)$ is a ranking function

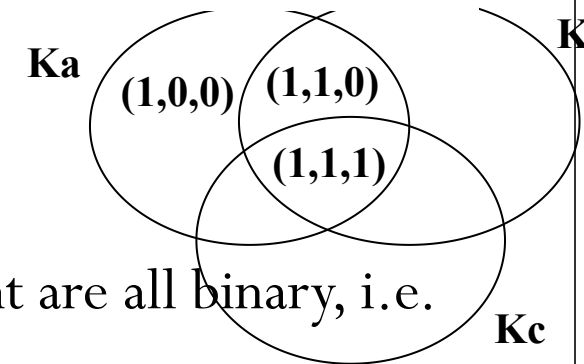




- Basic concepts: Each document is described by a set of representative keywords called *index terms*
- An index term is a word that represents a documents main theme.
- It is used to index and summarize the document contents.
- Assign a numerical *weights* to distinct relevance between index terms
- Three classic models: Boolean, vector, probabilistic
- k_i : A generic index term
- K : The set of all index terms $\{k_1, \dots, k_t\}$
- $w_{i,j}$: A weight associated with index term k_i of a document d_j
For an index term does not appear in the document , $w_{i,j}=0$
- g_i : A function returns the weight associated with k_i in any t -dimensional vector ($g_i(d_j)=w_{i,j}$)

Boolean model

- Simple retrieval model based on set theory and Boolean algebra
- Binary decision criterion
 - Either relevant or not relevant (no partial match)
- Data retrieval model
- Advantage
 - Simplicity
- Disadvantage
 - It is not simple to translate an information need into a Boolean expression
 - Exact matching may lead to retrieval of too few or too many documents



- Formal definition

- For the Boolean model, the index term weight are all binary, i.e. $w_{ij} \in \{0,1\}$
- A query q is composed of index terms linked by three connectives: not, and , or.

- A query is a conventional Boolean expression, which can be represented as a disjunction of conjunctive vectors (in DNF)
- $q = k_a \wedge (k_b \vee \neg k_c) = \vec{q}_{dnf} (1,1,1) \vee (1,1,0) \vee (1,0,0)$ where each of the component is a binary weighted vector associated with the tuple (k_a, k_b, k_c)

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

$$sim(d_j, q) = \begin{cases} 1 & \text{if } (\exists q_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, w_{i,j} = g_i(q_{cc})) \\ 0 & \end{cases}$$



A T M E
College of Engineering



- If $\text{sim}(d_j, q) = 1$ then the boolean model predicts that the document d_j is relevant to the query q
- Otherwise the prediction is that the document is not relevant.

Vector model

- Assign non-binary weights to index terms in queries and in documents
- Compute the similarity between documents and query = $\text{Sim}(d_j, q)$
- More precise than Boolean model
- The weight associated with the pair (k_i, d_j) positive and non-binary.
- The index terms in the query are also weighted
- The vector for a document d_j is represented by

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

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

- $\vec{q}$ is the query vector where $w_{i,q}$ be the weight associated with the pair (k_i, q) where $w_{i,q} \geq 0$
- Term weights are used to compute the degree of similarity between documents and the user query. It varies from 0 to +1.
- Instead of predicting a document relevant or not, it ranks the document according to the degree of similarity

- A document might be retrieved even if it matches the query only partially.
- Degree of similarity

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

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

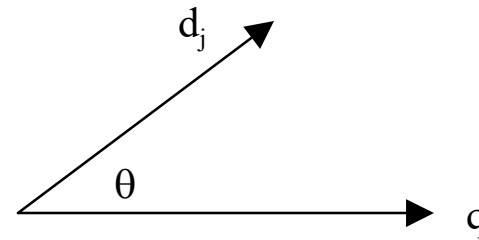


Figure 2.4 The cosine of θ is adopted as $sim(d_j, q)$

- Definition

- normalized frequency

$$f_{i,j} = \frac{freq_{i,j}}{\max_i freq_{i,j}}$$

- inverse document frequency $idf_i = \log \frac{N}{n_i}$

- term-weighting schemes $w_{i,j} = freq_{i,j} \times idf_i$

- query-term weights

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



- 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
- Disadvantage
 - The assumption of mutual independence between index terms

Probabilistic model

- Introduced by Roberston and Sparck Jones, 1976
 - Binary independence retrieval (BIR) model
- Idea: Given a user query q , and the ideal answer set R of the relevant documents, the problem is to specify the **properties** for this set
 - *Assumption (probabilistic principle): the probability of relevance depends on the query and document representations **only**; ideal answer set R should maximize the overall probability of relevance*
 - The probabilistic model tries to estimate the probability that the user will find the document d_j relevant with ratio
$$P(d_j \text{ relevant to } q) / P(d_j \text{ nonrelevant to } q)$$



- Definition

- All index term weights are all binary i.e., $w_{i,j} \in \{0,1\}$
- Let R be the set of documents known to be relevant to query q
- Let $\bar{R}$ be the complement of R
- Let $P(R | d_j)$ be the probability that the document d_j is relevant to the query q
- Let $P(\bar{R} | d_j)$ be the probability that the document d_j is nonelevant to query q

- The similarity $\text{sim}(\vec{d}_j, q)$ of the document $\vec{d}_j$ to the query q is defined as the ratio

$$\text{sim}(\vec{d}_j, q) = \frac{\Pr(R | \vec{d}_j)}{\Pr(\bar{R} | \vec{d}_j)}$$

- Using Bayes' rule,

$$\text{sim}(\vec{d}_j, q) = \frac{\Pr(\vec{d}_j | R) \times \Pr(R)}{\Pr(\vec{d}_j | \bar{R}) \times \Pr(\bar{R})}$$

- $P(R)$ stands for the probability that a document randomly selected from the entire collection is relevant
- $P(d_j | R)$ stands for the probability of randomly selecting the document $\vec{d}_j$ from the set R of relevant documents

$$\text{sim}(\vec{d}_j, q) \approx \log \frac{\Pr(\vec{d}_j | R)}{\Pr(\vec{d}_j | \bar{R})} + \log \frac{\Pr(R)}{\Pr(\bar{R})}$$

- Assuming independence of index terms and given $q = (d_1, d_2, \dots, d_t)$,

$$\Pr(\vec{d}_j | R) = \prod_{i=1}^t \Pr(k_i = d_i | R)$$

$$\Pr(\vec{d}_j | \bar{R}) = \prod_{i=1}^t \Pr(k_i = d_i | \bar{R})$$

$$\text{sim}(\vec{d}_j, q) \approx \log \frac{\prod_{i=1}^t \Pr(k_i = d_i | R)}{\prod_{i=1}^t \Pr(k_i = d_i | \bar{R})}$$



- $\Pr(k_i | R)$ stands for the probability that the index term k_i is **present** in a document randomly selected from the set R
- $\Pr(\bar{k}_i | R)$ stands for the probability that the index term k_i is **not present** in a document randomly selected from the set R

$$\text{sim}(\bar{d}_j, q) \approx \frac{\prod_{g_i(d_j)=1} \Pr(k_i | R) \prod_{g_i(d_j)=0} \Pr(\bar{k}_i | R)}{\prod_{g_i(d_j)=1} \Pr(k_i | \bar{R}) \prod_{g_i(d_j)=0} \Pr(\bar{k}_i | \bar{R})}$$

$$\because \Pr(k_i | R) + \Pr(\bar{k}_i | R) = 1$$

$$\text{sim}(\bar{d}_j, q) \approx \sum_{i=1}^t \left(\log \frac{P(k_i | R)}{1 - P(k_i | R)} + \log \frac{1 - P(k_i | \bar{R})}{P(k_i | \bar{R})} \right)$$

$$\text{sim}(\bar{d}_j, q) \approx \sum_{i=1}^t w_{i,q} \times w_{i,j} \times \left(\log \frac{P(k_i | R)}{1 - P(k_i | R)} + \log \frac{1 - P(k_i | \bar{R})}{P(k_i | \bar{R})} \right)$$