A document d is relevant to a query q, if $d \models q$

If a document does not support the query q, it does not necessarily mean that the document is not relevant to the query. Additional information, such as synonymys, hypernyms/hyponyms, meronyms, etc., can be used to transform the document d into d' such that $d' \models q$. Semantic relationships in a thesaurus, like WordNet, are useful sources for this information. The transformation from d to d' is regarded as flow of information between situations.

The *interaction IR* model was first introduced in Dominich (1992, 1993) and Rijsbergen (1996). In this model, the documents are not isolated; instead, they are interconnected. The query interacts with the interconnected documents. Retrieval is conceived as a result of this interaction. This view of interaction is taken from the concept of interaction as realized in the Copenhagen interpretation of quantum mechanics. Artificial neural networks can be used to implement this model. Each document is modelled as a neuron, the document set as a whole forms a neural network. The query is also modelled as a neuron and integrated into the network. Because of this integration, new connections are built between the query and the documents, and existing connections are changed. This restructuring corresponds to the concept of interaction. A measure of this interaction is obtained and used for retrieval. Detailed mathematical treatments of the model have been discussed by Dominich (1992, 1993, 2001) and van Rijsbergen (1996).

9.6 ALTERNATIVE MODELS OF IR

9.6.1 Cluster Model

The cluster model is an attempt to reduce the number of matches during retrieval. The need for clustering was first pointed out by Salton. Before we discuss the cluster-based IR model, we would like to state the cluster hypothesis that explains why clustering could prove efficient in IR.

Closely associated documents tend to be relevant to the same clusters.

This hypothesis suggests that closely associated documents are likely to be retrieved together. This means that by forming groups (classes or clusters) of related documents, the search time reduced considerably. Instead of matching the query with every document in the collection, it is matched with representatives of the class, and only documents from a class whose representative is close to query, are considered for individual match.

Clustering can be applied on terms instead of documents. Thus, terms can be grouped to form classes of co-occurrence terms. Co-occurrence terms can be used in dimensionality reduction or thesaurus construction. A number of methods are used to group documents. We discuss here, a cluster generation method based on similarity matrix. This method works as follows:

Let $D = \{d_1, d_2, \ldots, d_j, \ldots, d_m\}$ be a finite set of documents, and let $E = (e_{ij})_{n,n}$ be the similarity matrix. The element E_{ij} in this matrix, denotes a similarity between document d_i and d_j. Let T be the threshold value. Any pair of documents d_i and d_j $(i \neq j)$ whose similarity measure exceeds the threshold $(e_{ij} \geq T)$ is grouped to form a cluster. The remaining documents form a single cluster. The set of clusters thus obtained is

$$C = \{C_1, C_2, \ldots, C_k, \ldots, C_p\}$$

A representative vector of each class (cluster) is constructed by computing the centroid of the document vectors belonging to that class. Representation vector for a cluster C_k is

$$r_k = \{a_{1k}, a_{2k}, \ldots, a_{ik}, \ldots, a_{mk}\}$$

An element a_{ik} in this vector is computed as

$$a_{ik} = \frac{\sum_{d_j \in C_k} a_{ij}}{|C_k|}$$

where a_{ij} is weight of the term t_i, of the document d_j, in cluster C_k. During retrieval, the query is compared with the cluster vectors

$$(r_1, r_2, \ldots, r_k, \ldots, r_p)$$

This comparison is carried out by computing the similarity between the query vector q and the representative vector r_k as

$$s_{ik} = \sum_{i=1}^{m} a_{ik} q_i, \quad k = 1, 2, \ldots, p$$

A cluster C_k whose similarity s_k exceeds a threshold is returned and the search proceeds in that cluster.

Example 9.4

Let

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

be the term-by-document matrix. The similarity matrix corresponding to these documents is

1.0
0.9 1.0
0.4 0.4 1.0

Using a threshold of 0.7, we get the following two clusters:

$C_1 = \{d_1, d_2\}$
$C_2 = \{d_3\}$

The cluster vectors (representatives) for C_1 and C_2 are

$r_1 = (1\ 0.5\ 1\ 0\ 1)$
$r_2 = (0\ 0\ \ 1\ 1\ 0)$

Retrieval is performed by matching the query vector with r_1 and r_2.

9.6.2 Fuzzy Model

In the fuzzy model, the document is represented as a fuzzy set of terms, i.e., a set of pairs $[t_i, \mu(t_i)]$, where μ is the membership function. The membership function assigns to each term of the document a numeric membership degree. The membership degree expresses the significance of term to the information contained in the document. Usually, the significance values (weights) are assigned based on the number of occurrences of the term in the document and in the entire document collection, as discussed earlier. Each document in the collection

$$D = \{d_1, d_2, \ldots, d_j, \ldots, d_n\}$$

can thus be represented as a vector of term weights, as in the following vector space model

$$(w_{1j}, w_{2j}, w_{3j}, \ldots, w_{ij}, \ldots, w_{mj})^t$$

where w_{ij} is the degree to which term t_i belongs to document d_j.

Each term in the document is considered a representative of a subject area and w_{ij} is the membership function of document d_j to the subject area represented by term t_i. Each term t_i is itself represented by a fuzzy set f_i in the domain of documents given by

$$f_i = \{(d_j, w_{ij})\} \mid i = 1, \ldots, m;\ j = 1, \ldots, n$$

This weighted representation makes it possible to rank the retrieved documents in decreasing order of their relevance to the user's query.

Typically, queries are Boolean queries. For each term that appears in the query, a set of documents is retrieved. Fuzzy set operators are then applied to obtain the desired result.

For a single-term query $q = t_q$, those documents from the fuzzy set $f_q = \{(d_j, w_{iq})\}$, are retrieved for which w_{iq} exceeds a given threshold. The threshold may also be zero.

Consider the case of an AND query $q = t_{q1} \wedge t_{q2}$

First, the fuzzy sets f_{q1} and f_{q2} are obtained and then, their intersection is obtained, using the fuzzy intersection operator $f_{q1} \vee f_{q2} = \min \{(d_j, w_{iq1}), (d_j, w_{iq})\}$

The documents in this set are returned.

Similarly, for an OR query $q = t_{q1} \wedge t_{q2}$, the union of fuzzy sets f_{q1} and f_{q2} is computed to retrieve documents as follows:

$$f_{q1} \vee f_{q2} = \max \{(d_j, w_{iq1}), (d_j, w_{iq})\}$$

Example 9.5 Consider the following three documents:

d_1 = {information, retrieval, query}
d_2 = {retrieval, query, model}
d_3 = {information, retrieval}

where the set of terms used to represent documents is

T = {information, model, query, retrieval}

The fuzzy sets induced by these terms are

$f_1 = \{(d_1, 1/3), (d_2, 0) (d_3, 1/2)\}$
$f_2 = \{(d_1, 0), (d_2, 1/3) (d_3, 0)\}$
$f_3 = \{(d_1, 1/3), (d_2, 1/3) (d_3, 0)\}$
$f_4 = \{(d_1, 1/3), (d_2, 1/3) (d_3, 1/2)\}$

If the query is $q = t_2 \wedge t_4$, then document d_2 will be returned.

9.6.3 Latent Semantic Indexing Model

Latent semantic indexing model is the application of single value decomposition to IR. The use of latent semantic indexing (LSI) is based on the assumption that there is some underlying 'hidden' semantic structure in the pattern of word-usage across documents, rather than just surface level word choice. LSI attempts to identify this hidden semantic structure through statistical techniques and use it to represent and retrieve information. This is done by modelling the association between terms and documents based on the manner in which terms co-occur across documents. LSI transforms the term-document vector space into a more compact latent semantic space. Each dimension in the reduced space corresponds to an 'artificial concept'. These concepts loosely correspond to a set of terms. It is believed that in the vector space of reduced dimensionality, the words referring to related concepts, i.e., words that

co-occur, are collapsed into the same dimension. Latent semantic space is thus able to capture similarities that go beyond term similarity. In the latent semantic space, a query and a document can have high similarity even if the document does not contain a query term, provided the terms are semantically related.

Now we discuss how the LSI technique is actually employed in IR. The document collection is first processed to get a $m \times n$ term-by-document matrix, W, where m is the number of index terms and n is the total number of documents in the collection. Columns in this matrix represent document vectors, whereas the rows denote term vectors. The matrix element W_{ij} represents the weight of the term i in document j. The weight may be assigned based on term frequency or some combination of local and global weighting, as in the case of vector space model. Singular value decomposition (SVD) of the term-by-document matrix is then computed. Using SVD, the matrix is represented as a product of three matrices

$$W = TSD^T$$

where T corresponds to term vectors and has m rows and r columns and $r = \min(m, n)$. S corresponds to singular values. D^T is the transpose of D and has r rows and n columns. D corresponds to the document vector.

T and D are orthogonal matrices containing the left and right singular vectors of W. S is a diagonal matrix, containing singular values stored in decreasing order. We eliminate small singular values and approximate the original term-by-document matrix using truncated SVD. For example, by considering only the first k number of the largest singular values, along with their corresponding columns in T and D, we get the following approximation of the original term-by-document matrix in a space of k orthogonal dimensions, where k is sufficiently less than n:

$$W_k = T_k S_k D_k^T$$

where T_k is the first k columns of T, D_k^T is the first k columns of D^T, and S_k is the k largest singular values.

The matrix W_k is used for retrieval. The idea is that the elimination of small singular values throws out the 'noise' resulting from term usage variation, and captures the underlying 'hidden' semantic structure (i.e., concepts). Each dimension in the reduced space corresponds to artificial or derived concepts. Each such concept loosely represents a set of terms in the original term-document matrix. Documents with varying word usage patterns are collapsed to the same vector in k-space.

The queries are also represented in k-dimensional space. Let $q = (q_1, q_2, \ldots, q_m)$ be the original query vector, where each element q_i is the frequency

of term i in the query q. The query q is represented in the k-dimensional space as

$$q_k = q^T T_k S_k^{-1}$$

where q^T is the transpose of the query vector, and T_k and S_k are the weights. $q^T T_k$ denotes the sum of k-dimensional term vectors and S_k^{-1}, the weights of each dimension. Thus, the query is represented as the weighted sum of its constituent term vectors.

Retrieval is performed by computing the similarity between query vector and document vector. For example, we can use the cosine similarity measure to rank documents to perform retrieval. In a keyword-based retrieval, relevant documents that do not share any term with the query are not retrieved. The LSI-based approach is capable of retrieving such documents, as similarity is computed based on the overall pattern of term usage across the document collection rather than on term overlap.

We now give an example to explain how a document in high-dimensional space is represented in a low, reduced, latent semantic space.

Example 9.6 Consider the matrix shown in Figure 9.5. This matrix defines five-dimensional space in which six documents, d_1, d_2, d_3, ..., d_6, have been represented. The five dimensions correspond to five index terms *tornado, storm, tree, forest,* and *farming*. For simplicity, tf has been used to weight index terms. Figure 9.6 shows the documents in a two-dimensional space. The vectors in the figure correspond to document vectors in the matrix R, which is the representation of X in reduced two-dimensional space. The two dimensions correspond to derived concepts obtained through the application of truncated SVD.

$$X = \begin{array}{l|cccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{tornado} & 1 & 1 & 0 & 0 & 0 & 0 \\ \text{storm} & 1 & 0 & 1 & 0 & 1 & 0 \\ \text{tree} & 1 & 0 & 1 & 0 & 0 & 0 \\ \text{forest} & 0 & 0 & 1 & 1 & 0 & 0 \\ \text{farming} & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Figure 9.5 A term-document matrix representing six documents in five-dimensional space

We now explain how to arrive at the reduced dimensionality representation of X. First, the SVD of X is computed to get the three matrices T, S, and D.

$$X_{5\times 6} = T_{5\times 5}\, S_{5\times 5}\, (D_{6\times 5})^T$$

These matrices are shown in Figures 9.7, 9.8, and 9.9 respectively. Consider the first two largest singular values of S, and rescale $D^T_{2\times 6}$ with singular

values to get matrix $R_{2\times6} = S_{2\times2}D^T{}_{2\times6}$, as shown in Figure 9.10, where $S_{2\times2}$ is S restricted to two dimensions and $D^T_{2\times6}$ is D^T restricted to two columns. R is a reduced dimensionality representation of the original term-by-document matrix X and is used to plot the vectors in Figure 9.6.

To find out the changes introduced by the reduction, we compute document similarities in the new space and compare them with the similarities between documents in the original space. The document–document correlation matrix for the original n-dimensional space is given by the matrix $Y = X^TX$. Here, Y is a square, symmetric $n \times n$ matrix. An element Y_{ij} in this matrix gives the similarity between documents i and j. The correlation matrix for the original document vectors is shown in Figure 9.12. This matrix is computed using X, after normalizing the lengths of its columns. The document–document correlation matrix for the new space is computed analogously using the reduced representation R. Let N be the matrix R with length-normalized columns. Then, $M = N^TN$ gives the matrix of document correlations in the reduced space. The correlation matrix M is given in Figure 9.11. The similarity between document d_1, d_4(–0.0304), and d_6(–0.2322) is quite low in the new space because document d_1 is not topically similar to documents d_4 and d_6. In the original space, the similarity between documents d_2 and d_3 and between documents d_2 and d_5 is 0. In the new space, they have high similarity values (0.5557 and 0.8518 respectively) although documents d_3 and d_5 share no term with the document d_2. This topical similarity is recognized due to the co-occurrence of patterns in the documents.

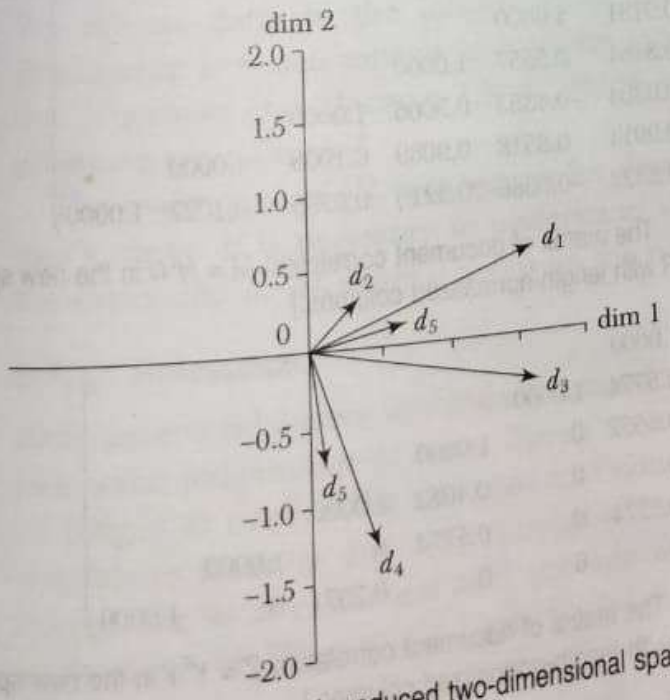

Figure 9.6 Documents in reduced two-dimensional space

$$T = \begin{pmatrix} 0.3318 & 0.3338 & 0.8064 & -0.2426 & -0.2634 \\ 0.6693 & 0.1616 & -0.2737 & 0.5853 & -0.3293 \\ 0.5514 & 0.1038 & -0.0961 & -0.2667 & 0.7777 \\ 0.3583 & -0.5745 & -0.2148 & -0.5778 & -0.4021 \\ 0.0974 & -0.7223 & 0.4684 & 0.4400 & 0.2362 \end{pmatrix}$$

Figure 9.7 Matrix T for the SVD of the term-document matrix X shown in Figure 9.5

$$S = \begin{pmatrix} 2.3830 & 0 & 0 & 0 & 0 \\ 0 & 1.6719 & 0 & 0 & 0 \\ 0 & 0 & 1.2415 & 0 & 0 \\ 0 & 0 & 0 & 0.8288 & 0 \\ 0 & 0 & 0 & 0 & 0.5454 \end{pmatrix}$$

Figure 9.8 The matrix S for singular values of the SVD of the term-document matrix X

$$D^T = \begin{pmatrix} 0.6515 & 0.1392 & 0.6626 & 0.1912 & 0.2809 & 0.0409 \\ 0.3584 & 0.1996 & -0.1848 & -0.7756 & 0.0967 & -0.4320 \\ 0.3516 & 0.6495 & -0.4710 & 0.2042 & -0.2205 & 0.3773 \\ 0.0916 & -0.2927 & -0.3127 & -0.1662 & 0.7062 & 0.5309 \\ 0.3392 & -0.4829 & 0.0849 & -0.3042 & -0.6037 & 0.4330 \end{pmatrix}$$

Figure 9.9 The matrix D^T for singular values of the SVD of the term-document matrix

$$R = \begin{pmatrix} 1.5526 & 0.3318 & 1.5790 & 0.4557 & 0.6693 & 0.0974 \\ 0.5992 & 0.3338 & -0.3090 & -1.2967 & 0.1616 & -0.7223 \end{pmatrix}$$

Figure 9.10 The matrix $R_{2\times 6} = S_{2\times 2} D^T_{2\times 6}$ representing documents in two-dimensional space

$$M = \begin{pmatrix} 1.0000 & & & & & \\ 0.9131 & 1.0000 & & & & \\ 0.8464 & 0.5557 & 1.0000 & & & \\ -0.0304 & -0.4353 & 0.5066 & 1.0000 & & \\ 0.9914 & 0.8518 & 0.9089 & 0.1008 & 1.0000 & \\ -0.2322 & -0.6086 & 0.3215 & 0.9793 & -0.1027 & 1.0000 \end{pmatrix}$$

Figure 9.11 The matrix of document correlation $M = N^T N$ in the new space (N is matrix R with length-normalized columns.)

$$Z = \begin{pmatrix} 1.0000 & & & & & \\ 0.5774 & 1.0000 & & & & \\ 0.6667 & 0 & 1.0000 & & & \\ 0 & 0 & 0.4082 & 1.0000 & & \\ 0.5774 & 0 & 0.5774 & 0 & 1.0000 & \\ 0 & 0 & 0 & 0.7071 & 0 & 1.0000 \end{pmatrix}$$

Figure 9.12 The matrix of document correlation $Z = Y^T Y$ in the new space (Y is matrix X with length-normalized columns.)