Module 2

CHAPTER 2 Modelling in Information Retrieval

Syllabus

Taxonomy of Information Retrieval models, Classic Information Retrieval, Alternate set: Theoretical model, Alternative Algebraic models, Alternative Probabilistic models.
Structured text Retrieval models, Models for browsing.

2.1 TAXONOMY OF INFORMATION RETRIEVAL MODELS

GQ. Explain the taxonomy of information retrieval models in details. **(4 Marks)**

- An Information Retrieval model is defined as a quadruple [D, Q, F, Rel(q_i,d_j)] where
  - D represents a group of documents found in a collection.
  - Q represents a group of information needs which is referred to as queries.
  - F represents a Framework that consists of documents, queries with their relationships with those documents.
  - Rel ($q_{i,dj}$) represents the score which is associated with the query q_i and the document d_j. It is used to arrange the documents in order to be displayed to the user.
- In order to build a model, you must first understand how the document and query are represented.
- The framework is built on top of this representation. It also allows the documents to be ordered based on the score generated, which represents the level of relevance between the query and the document.
- The documents and basic set operations are available in the framework for the traditional Boolean model, the algebra operations on vectors are available in the framework for the vector model, and the Bayes theorem and probability operations are available in the framework for the probabilistic model.
- As time passed, various other alternative models for each type of classic model were proposed.
- Alternative set-theoretic models include the extended Boolean model and the fuzzy model; alternative algebraic models include the neural network model, latent semantic indexing, and generalised vector models; and alternative probabilistic models include the inference network and the belief network.
- Beyond referring to the textual content in the documents, structural models such as the proximal nodes model and the Non-overlapping lists model are used to refer to the structure present in the written text.

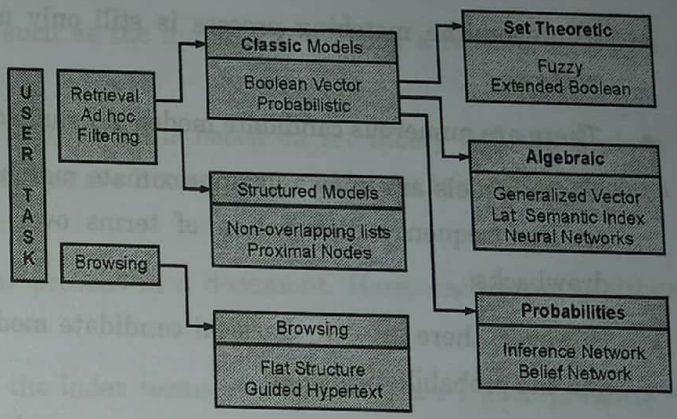


Fig. 2.1.1 : Information Retrieval Models

2.2 CLASSIC INFORMATION RETRIEVAL

GQ. Explain the classic model of IR. **(4 Marks)**
GQ. Explain the working of Boolean model. **(6 Marks)**
GQ. Discuss with suitable example, working of Boolean model. **(6 Marks)**
GQ. Explain the vector space model.
GQ. What is vector space model? Explain with suitable diagram. **(6 Marks)**
GQ. Explain the terms: tf, idf, degree of similarity. **(4 Marks)**
GQ. List and explain various additional IR models. **(6 Marks)**

The classical information retrieval models are depicted in Fig. 2.1.1. The following sections elaborate o the classic models.

Basic models of information retrieval a brief overview

- A mathematical model of information retrieval guides the implementation of information retrieva systems.
- Only the matching process is automated in traditional information retrieval systems, which ar typically operated by professional searchers; indexing and query formulation are manual processes.
- Mathematical models of information retrieval must thus only model the matching process for these systems.
- The Boolean model of information retrieval is used in practise by traditional information retrieva systems.

2.2.1 The Boolean Model

- Is an exact matching model, which means it either retrieves or does not retrieve documents without ranking them?
- The model allows for the use of structured queries, which include not only query terms but also relationships between the terms defined by the query operators AND, OR, and NOT.
- Query formulation is also automated in modern information retrieval systems, which are typically operated by nonprofessional users.
- However, the matching process is still only modelled in candidate mathematical models for these systems.
- There are numerous candidate models for ranked retrieval systems' matching process.
- These models are known as approximate matching models because they rank the retrieved sets based on the frequency distribution of terms over documents. Each of these models has benefits and drawbacks.
- However, there are two classical candidate models for approximate matching: the vector space model and the probabilistic model.
- They are classical models, not only because they were introduced already in the early 70's, but also because they represent classical problems in information retrieval.
- This model is regarded as one of the oldest and most traditional information retrieval models.
- This model is well explained by mapping the query terms to a set of documents. For example, the term "Botany" defines and indexes all documents containing the term "Botany."
- The terms in the query and the documents in question can be combined using the Boolean operators to create an entirely new set of documents.
- When the AND operator is used between two terms, it returns a set of documents that are smaller or equal to the document set otherwise, whereas the OR operator returns a set of documents that are

greater or equal to the document set otherwise.

- For example, the information need "Botany AND Zoology" will return a set of documents containing both words, whereas the query with the keywords "Botany AND Zoology" will return a set of documents containing either the word "Botany" or "Zoology."
- The following Venn diagrams clearly explain the representation (Fig. 2.2.1, Fig. 2.2.2 and Fig. 2.2.3). Grey areas represent the set of documents that can be retrieved.

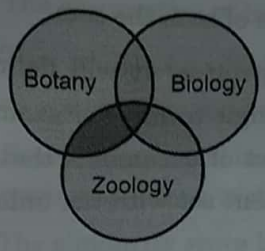


Fig. 2.2.1 : Botany AND Zoology

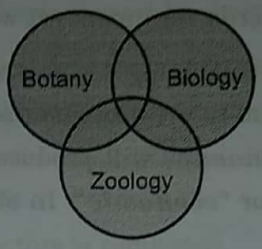


Fig. 2.2.2 : Botany OR Zoology

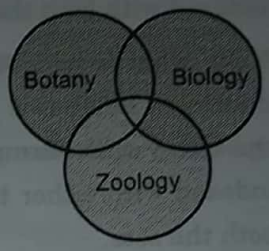


Fig. 2.2.3 : (Botany OR Biology) AND NOT (Botany AND Zoology)

- This model gives the user a sense of system control. It is because the end user immediately understands whether or not the document is retrieved. It is also simple to understand why the document is or is not retrieved.
- In the case of a query "Botany AND Zoology AND Biology," it will not return a document that contains the terms "Family," "Friends," or "Parents," but it will also return a document that contains "Botany" and "Zoology" but not "Biology."
- However, this model has significant limitations, such as the inability to provide a ranking based on relevance when retrieving multiple documents.
- It is the oldest information retrieval (IR) model. The model is based on set theory and the Boolean algebra, where documents are sets of terms and queries are Boolean expressions on terms. The Boolean model can be defined as
  - **D** – A set of words, i.e., the indexing terms present in a document. Here, each term is either present (1) or absent (0).
  - **Q** – A Boolean expression, where terms are the index terms and operators are logical products – AND, logical sum – OR and logical difference – NOT.
  - **F** – Boolean algebra over sets of terms as well as over sets of documents.

  If we talk about the relevance feedback, then in Boolean IR model the Relevance prediction can be defined as follows :
  - **R** – A document is predicted as relevant to the query expression if and only if it satisfies the query expression as :

  ((text ∨ information) ∧retrieval ∧~ theory)

We can explain this model by a query term as an unambiguous definition of a set of documents.

- For example, the query term ***"economic"*** defines the set of documents that are indexed with the t ***"economic"***.
- Now, what would be the result after combining terms with Boolean AND Operator? It will defin document set that is smaller than or equal to the document sets of any of the single terms. For exam the query with terms ***"social"*** and ***"economic"*** will produce the documents set of documents that indexed with both the terms. In other words, document set with the intersection of both the sets.
- Now, what would be the result after combining terms with Boolean OR operator? It will defin document set that is bigger than or equal to the document sets of any of the single terms. For exam the query with terms ***"social"*** or ***"economic"*** will produce the documents set of documents that indexed with either the term ***"social"*** or ***"economic"***. In other words, document set with the union both the sets.

☞ **Advantages of the Boolean Mode**

The advantages of the Boolean model are as follows :

(i) The simplest model, which is based on sets.

(ii) Easy to understand and implement.

(iii) It only retrieves exact matches.

(iv) It gives the user, a sense of control over the system.

☞ **Disadvantages of the Boolean Model**

The disadvantages of the Boolean model are as follows :

(i) The model's similarity function is Boolean. Hence, there would be no partial matches. This can b annoying for the users.

(ii) In this model, the Boolean operator usage has much more influence than a critical word.

(iii) The query language is expressive, but it is complicated too.

(iv) No ranking for retrieved documents.

2.2.2 Vector Space Model

- The problem of ranking the documents given the initial query is represented.
- The Vector model, which is likely the most popular, assigns real non-negative weights to index terms in documents and queries.
- Documents are represented in this model as vectors in a multidimensional Euclidean space.
- Each dimension in this space corresponds to a relevant term/word in the collection of documents.
- The degree of similarity of documents to queries is measured as the correlation between the vectors representing the document and the query, which can and is typically quantified by the cosine of the angle between the two vectors.

- In the vector model, index term weights are typically computed as a function of two factors: the term frequency factor, TF, a measure of intra-cluster similarity; computed as the number of times the term occurs in the document, normalised so that it is independent of document length; and an inverse document frequency, IDF, a measure of inter-cluster dissimilarity; weights each term based on its discriminative power across the entire collection.
- The main advantages of this model are related to improved retrieval performance due to term weighting and partial matching, which allows retrieval of documents that approximate the query conditions.
- The assumption of index term independence is most likely its main disadvantage.
- This model is a widely used model proposed by Gerard Salton and his team of researchers that is based on a similarity condition explained by the vector representation.
- Every document in the document space and every information need expressed as a query are represented by a vector in the term space.
- The similarity score between the two vectors is computed.
- This model considers the notion that the document is expressed using a set of words, and thus the words represented in a vector can be considered the document representation.
- The query, which is a collection of keywords, can also be represented as a vector.
- The similarity score can be calculated by measuring the distance between the document vector and the query vector, which represents how closely or distantly related the document is to the query.
- The Fig. 2.2.4 depicts the query and document representations in the vector space model. The vector representation of a document and query that is spanned by the terms Botany, Zoology, and Biology is shown in Fig. 2.2.4.

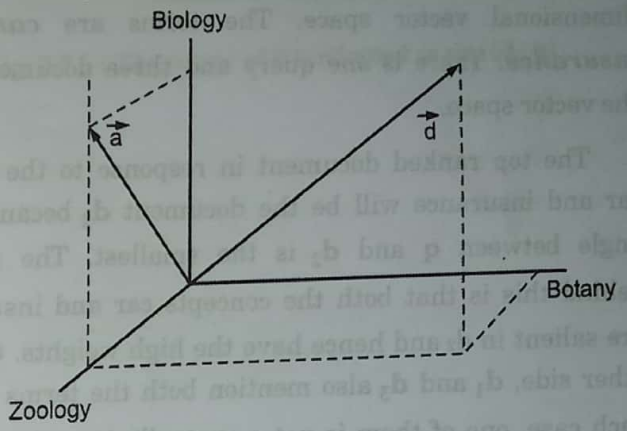


Fig. 2.2.4 : A query and documents representation in the vector space model

The similarity score is normally the cosine of the angle that separates the two vectors

→ q and

→ d. The cosine of the angle is 0 if the vectors are orthogonal in the space and is 1 if the angle is 0 degrees. The formula is given as follows :

Score (
→d,
→q) = ...(1)

$$\sum k = 1mn\,(dk) \cdot n\,(qk) \quad ...(2)$$

Where $n\,(vk) = vk \sum k = 1\, mvK^2$

- The representation of angles between vectors in dimensional space simplifies explanation.
- This geometric interpretation, adapted in the vector space approach, makes it simple to use information retrieval challenges.
- It is also widely used in the fields of document clustering and automatic categorization of textual data

Due to the above disadvantages of the Boolean model, Gerard Salton and his colleagues suggested model, which is based on Luhn's similarity criterion. The similarity criterion formulated by Luhn stat "the more two representations agreed in given elements and their distribution, the higher would be t probability of their representing similar information."

Consider the following important points to understand more about the Vector Space Model :

- The index representations (documents) and the queries are considered as vectors embedded in a hig dimensional Euclidean space.
- The similarity measure of a document vector to a query vector is usually the cosine of the angle betwe them.

☞ Vector Space Representation with Query and Document

The query and documents are represented by a two-dimensional vector space. The terms are ***car*** and ***insurance***. There is one query and three documents in the vector space.

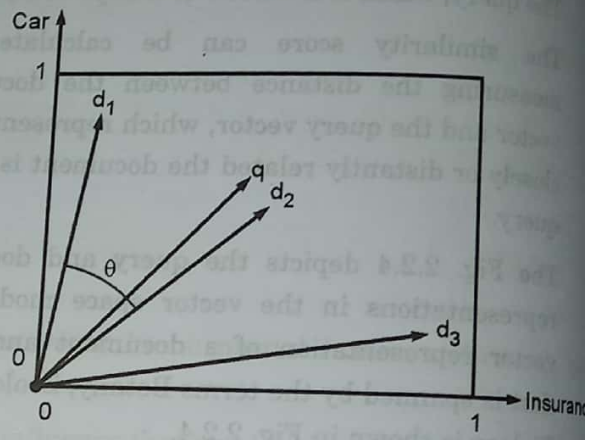


Fig. 2.2.5

The top ranked document in response to the terms car and insurance will be the document $\mathbf{d_2}$ because the angle between $\mathbf{q}$ and $\mathbf{d_2}$ is the smallest. The reason behind this is that both the concepts car and insurance are salient in d_2 and hence have the high weights. On the other side, $\mathbf{d_1}$ and $\mathbf{d_3}$ also mention both the terms but in each case, one of them is not a centrally important term in the document.

☞ Vector Model [Salton, 1968]

- *Assign non-binary weights to index terms in queries and in documents* → TFxIDF.
- *Compute the similarity between documents and query* → Sim (Dj, Q)
- *More precise than Boolean model*
- ***Idea for TFxIDF*** →
- ***TF :*** *intra-clustering similarity is quantified by measuring the raw frequency of a term* k_i *inside a document* d_j.
  - *Term frequency (the tf factor) provides one measure of how well that term describes the document contents.*

- **IDF :** Inter-clustering similarity is quantified by measuring the inverse of the frequency of a term k_i among the documents in the collection.
  - inverse document frequency (the idf factor)
- Index terms are assigned positive and non-binary weights.
- The index terms in the query are also weighted

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

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

- Term weights are used to compute the degree of similarity between documents and the user query.
- Then, retrieved documents are sorted in decreasing order .
- **Degree of similarity**

$$\text{sim}(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\left|\vec{d_j}\right| \times \left|\vec{q}\right|}$$

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

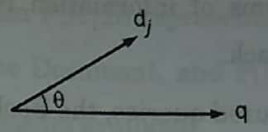


Fig. 2.2.6 : The cosine of θ is adopted as sim (d_j, q)

- **Definition**
- **Normalized frequency**

$$f_{i,j} = \frac{\text{freq}_{i,j}}{\max \text{freq}_{i,j}}$$

- Inverse document frequency

$$\text{idf}_i = \log \frac{N}{n_i}$$

- Term-weighting schemes

$$w_{i,j} = \text{freq}_{i,j} \times \text{idf}_i$$

- Query-term weights

$$w_{i,q} = \left(0.5 + \frac{0.5\ \text{freq}_{i,q}}{\max \text{freq}_{i,q}}\right) \times \log \frac{N}{n_i}$$

☞ **Advantages**

(i) Its term-weighting scheme improves retrieval performance.

(ii) Its partial matching strategy allows retrieval of documents that approximate the query conditions.

(iii) Its cosine ranking formula sorts the documents according to their degree of similarity to the query.

Disadvantages

The assumption of mutual independence between index terms.

2.2.3 Probabilistic Model

- Represent the problem of document ranking after some feedback has been gathered.
- The similarity between documents and queries is computed by probabilistic models as the odds of a document being relevant to a query.
- The weights of index terms are binary. This model ranks documents in decreasing order of likelihood of relevance, which is advantageous.
- Its main drawbacks are: the need to guess the initial separation of documents into relevant and non-relevant categories; binary weights; and index terms assumed to be independent.
- In practise, the Boolean model, the vector space model, and the probabilistic model represent three classical problems of information retrieval, namely structured queries, initial term weighting, and relevance feedback.
- To create structured queries, the Boolean model provides the query operators AND, OR, and NOT.
- If examples of relevant documents are available, the probabilistic model provides a theory of optimal ranking.
- As the name implies, this model is based on the Theory of Probability. As a result, the similarity score between the query and the document is computed with the probability that the document is relevant to the query.
- Based on the underlying model, numerous approaches were proposed. Consider the probability of relevance, denoted by P (R), and the set of all possible outcomes in the experiment, denoted by sample space.
- The outcome of P (R) will be either relevant or irrelevant, where "1" denotes relevant and "0" denotes irrelevant.
- If there are 2 million documents in a collection and 200 of them are relevant to the collection, the probability of relevance is calculated as follows.

  $P(R = 1) = 200/2{,}000{,}000 = 0.0001.$

- Assume P (Dt) is the probability that a document contains the term "t," and the sample space is represented as 0,1, where "0" is the value if the term "t" is not present in the document, and "1" is the value if the term "t" is present in the document.
- The probability P (R, Dt) represents the combined probability of several outcomes, which are represented as (0,0), (0,1), (1,0), (1,1). $P(R = 1 | Dt = 1)$ is the relevance probability if the document containing the terms "t" is considered.

- Over time, various models based on probability theory, such as the Probabilistic Indexing model and the Probabilistic Retrieval model, have been proposed and used to address various IR challenges. Fig. 2.2.7 shows an example of the Probabilistic Retrieval model in action.

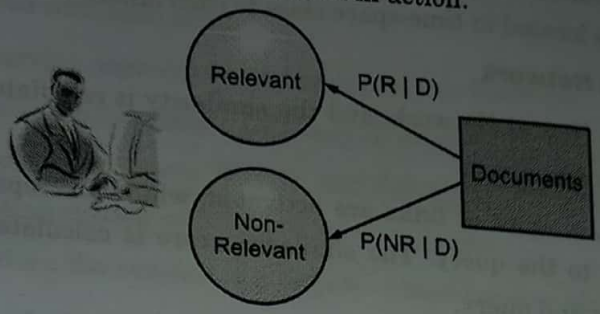


Fig. 2.2.7 : The probabilistic retrieval

- If 'R' represents the total number of Relevant documents, 'NR' represents the total number of Non Relevant documents, and 'D' represents the document space, then P(R|D) represents the probability of relevant documents among the total documents available in the Document, and P(NR |D) represents the probability of non-relevant documents among the total documents available.
- The Bayes Rule, which is also used in the Probability model, is defined as follows.

$$P(R \mid D) = P\,DRP(R)\,P(D) \quad \ldots(3)$$

- P(R|D) can be interpreted in the probability retrieval model as follows. If there are 20 documents represented by the letter 'D,' and 18 of them are relevant, then P(R|D) = 0.9. In section 1.3.3, a few more models that are used in many other areas of information retrieval are briefly explained.

☞ Additional Models

Various other models have also been specified over time, and they deal with different features. Several of them are briefly discussed below:

(a) Language Models

This model is predicated on the idea that each document is represented as a language model, and the likelihood of the document generating the information need is calculated.

(b) Model-based on Inference Network

This model employs a Bayesian Network to determine the relevance of the document to the imposed query. The "evidence" in the document about its relevance allows the inference to be made. The inference is used to compute the similarity score.

(c) Model-based on Latent Semantic Indexing

The Term-document matrix representation is used to represent the term occurrence in the docume The single value decomposition (SVD) is used to remove noise from the document so that many docume with similar semantics can be located in time-space close to each other.

(d) Model-based on Neural Network

- This model is based on a Neural Network, and the similarity is calculated based on the links betwe the query and the documents.
- When a query is entered, a series of links are activated, which are represented as Neurons or Nod that connect a document to the query. The similarity score is calculated using the number of nod available for the document and query.
- The network is then trained further by varying the weights in the links that connect the document the query.

(e) Model-based on Genetic Algorithm

- This model focuses on the concept of evolution, which implies that the optimised query aimed at findi relevant documents must evolve.
- The seed query, which is the first one generated, is expanded into a newer one with individual term weight estimates or random values.
- As the process progresses, the newly formed query survives because it is close to the set of relevant documents, and the other related queries that do not fit are later removed from further processing.

(f) Model-based on Fuzzy Set Retrieval

- This model is based on the concept of the Fuzzy set, in which the documents in the document set are mapped to a Fuzzy set that includes not only the elements but also a number that indicates the membership strength.
- The AND, OR, and NOT operators are used to construct Boolean queries that yield the strength of membership associated with each document that is related to the query. The similarity score is calculated using these values.

2.3 SET THEORETIC MODEL

- The Boolean model imposes a binary criterion for deciding relevance
- The question of how to extend the Boolean model to accommodate partial matching and a ranking has attracted considerable attention in the past
- Two set theoretic models are-
  1. Fuzzy Set Model
  2. Extended Boolean Model

2.3.1 Fuzzy Set Model

- Queries and documents are represented by sets of index terms, so matching is always approximate.
- This ambiguity can be modelled using a fuzzy framework, as follows :
  1. A fuzzy set is associated with each term.
  2. Each document has varying degrees of membership in this fuzzy set.
- This interpretation serves as the foundation for many IR models based on fuzzy theory.
- We discuss the model proposed by Ogawa, Morita, and Kobayashi in this section (1991).
- A framework for representing classes with ill-defined boundaries.
- The main idea is to introduce the concept of a degree of membership associated with the elements of a set.
- This degree of membership ranges from 0 to 1 and allows for the simulation of the concept of marginal membership.
- Thus, membership is now a gradual concept, as opposed to the crisp concept imposed by classic Boolean logic.
- **Fuzzy Set Theory**
  - A query term: a fuzzy set
  - A document: degree of membership in this test
  - **Membership Function**
    - Associate membership function with the elements (documents) of the class.
      - 0 : no membership in test
      - 1 : full membership
      - 0~1 : marginal elements of the test

a class for query term document collection

↓ ↓ ↓

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

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

  - **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 $\{\}$ respectively.

Assume $\mu_A = \{d_1{:}0.8, d_2{:}0.7, d_3{:}0.6, d_4{:}0, d_5{:}0, d_6{:}0\}$

and $\mu_B = \{d_1{:}0, d_2{:}0.6, d_3{:}0.8, d_4{:}0.0, d_5{:}0, d_6{:}0\}$

$\mu_{\overline{A}}(u) = 1 - \mu_A(u) = \{d_1 : 0.2, d_2 : 0.3, d_3 : 0.4, d_4 : 1, d_5 : 1, d_6 : 1\}$

$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u)) = \{d_1 : 0.8, d_2 : 0.7, d_3 : 0.8, d_4 : 9, d_5 : 0, d_6 : 0\}$

$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u)) = \{d_1 : 0, d_2 : 0.6, d_3 : 0.6, d_4 : 0, d_5 : 0, d_6 : 0\}$

Fuzzy Information Retrieval

- Fuzzy generalisations of the Boolean model are used in the fuzzy information retrieval model.
- The fuzzy information retrieval model specifies the ambiguous relationship between the query language and the documents that are returned.
- The fuzzy information retrieval system assumes that each query word is associated with a set of fuzzy documents.
- That is, each query language word defines a fuzzy set, and the elements in the sets are retrieved documents. To correspond to each word in the query language, each document in the set has a degree of membership.
- The fuzzy set represents how well each document matches the query as a retrieval result.
- Indexing is the first step in creating a representation of a document. During the indexing definition process, we must ensure that the indexing can present textual information not only accurately but also comprehensively.
- To calculate the correlation between words, an indexing function is used as the fuzzy set's membership function.
- In other words, the membership function results are the weights assigned to index items and words in retrieval documents.
- People can use the function to achieve the goal of properly presenting textual information.

Basic idea

- Expand the set of index terms in the query with related terms (from the thesaurus) so that more relevant documents can be retrieved.
- A thesaurus can be created by defining a term-term correlation matrix c, the rows and columns of which correspond to the index terms in the document collection.
- Normalized correlation factor $c_{i,1}$ between two terms k_i and k_1 (0~1).

$$C_{i,l} = \frac{n_{i,l}}{n_i + n_l - n_{i,l}}$$

Where, n_i is # of documents containing term $\mathbf{k_i}$

n_l is # of documents containing term $\mathbf{k_l}$

$n_{i,l}$ is # of documents containing $\mathbf{k_i}$ **and** $\mathbf{k_l}$

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

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

☞ **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_i of d_j which is strongly related to the index k_i, them $\mu_{i,j} \sim 1$.
- k_i is a good fuzzy index.
- When all index terms of d_j are only loosely related to k_i, $\mu_{i,j} \sim 0$.
- k_i is a not good fuzzy index.

Example :

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

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

$$= CC_1 + CC_2 + CC_3$$

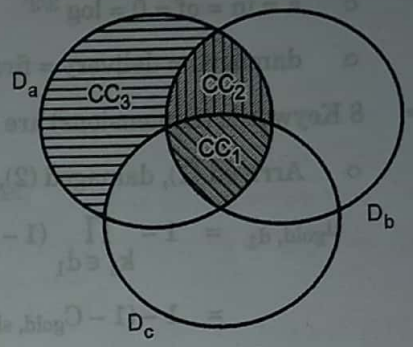


Fig. 2.3.1

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

$d_j \in D_a$ has a degree of membership.

μ_{aj} > a predefined threshold K

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

(the negation of index term k_a)

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

Disjunction normal form $\vec{q_{dnf}} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$

(1) The degree of membership in a disjunction fuzzy set is computed using an algebraic sum (instead of max function) more smoothly.

(2) The degree of membership in a conjunction fuzzy set is computed using an algebraic product (instead of min function).

Fuzzy Set Model

$$\mu_{q,j} = \mu_{cc_1 + cc_2 + cc_3, j} = 1 - \prod_{i=1}^{3} (1 - \mu_{cc_i, j})$$

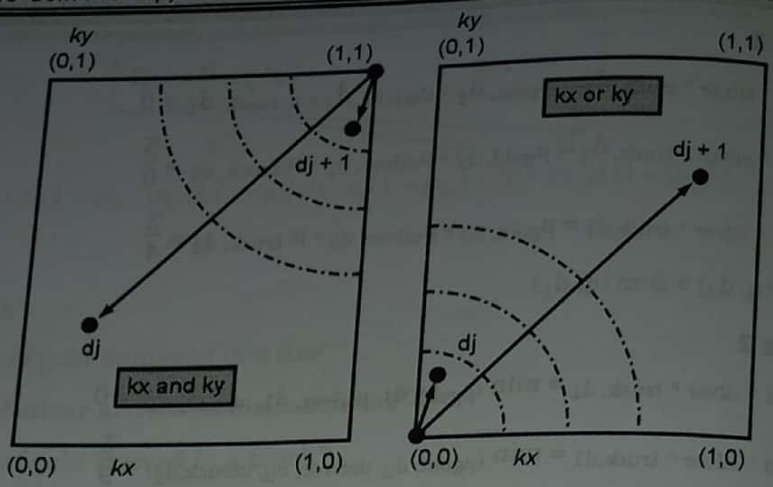


Fig. 2.3.1 : Extended Boolean logic considering the space composed of two terms k_x and k_y only

- For query $q = K_x$ or K_y, (0, 0) is the point we try to avoid. Thus, we can use, to rank to documents.

$$\text{Sim}(q_{or}, d) = \sqrt{\frac{x^2 + y^2}{2}}$$

- The bigger the better.
- For query $q = K_x$ or K_y, (1, 1) is the most desirable points.
- We use

$$\text{Sim}(q_{and}, d) = 1 - \sqrt{\frac{(1-x)^2 + (1-y)^2}{2}}$$

- The bigger the better.

☞ Extend the idea to m terms

$$q_{or} = k_1 \vee^p k_2 \vee^p \ldots\ldots \vee^p k_m$$

$$\text{Sim}(q_{or}, d_j) = \left(\frac{x_1^p + x_2^p + \ldots + x_m^p}{m}\right)^{1/p}$$

$$q_{and} = k_1 \vee^p k_2 \vee^p \ldots\ldots \vee^p k_m$$

$$\text{sim}(q_{and}, d_j) = 1 - \left(\frac{(1-x_1)^p + (1-x_2)^p + \ldots + (1-x_m)^p}{m}\right)^{1/p}$$

☞ Properties

The p norm as defined above enjoys a couple of interesting properties as follows. First, p = 1 it can verified that

$$\text{Sim}(q_{or}, d_j) = \text{Sim}(q_{and}, d_j) = \frac{x_1 + \ldots + x_m}{m}$$

Second, when $p = \infty$ it can be verified that

$$\text{Sim}(q_{or}, d_j) = \max(x_i)$$

$$\text{Sim}(q_{and}, d_j) = \min(x_i)$$

Example

For instance, consider the query $q = (k_1 \wedge k_2) \vee k_3$. The similarity sim (q, d_j) between a document dj and this query is then computed as

$$sim(q, d) = \left(\frac{\left(1 - \left(\frac{(1-x_1)^p + (1-x_2)^p}{2}\right)^{1/p}\right)^p + x_3^p}{2} \right)^{1/p}$$

Any Boolean can be expressed as a numeral formula.

2.4 ALTERNATIVE ALGEBRAIC MODELS

GQ. Explain the generalized vector model. (4 Marks)

2.4.1 Generalized Vector Model

- **Premise**
  - Classic models enforce independence of index terms.
  - **For the Vector model**
  - Set of term vectors $\{\vec{k}_1, \vec{k}_1, ..., \vec{k}_t\}$ are linearly independent and form a basis for the subspace of interest.
  - Frequently, it means pairwise orthogonality.

    $\forall_{i,j} \Rightarrow \vec{k}_i \cdot \vec{k}_j = 0$ (in a more restrictive sense)
  - An alternative interpretation: The index term vectors are linearly independent, but not pairwise orthogonal.
    - **Generalized Vector Model.**
- **Key idea**
  - Index term vectors form the basis of the space are not orthogonal and are represented in terms of smaller components (minterms).
- **Notations**
  - $\{k_1, k_2, \ldots, k_t\}$: the set of all terms
  - $w_{i,j}$: the weight associated with $[k_i, d_j]$
  - Minterms: binary indicators (0 or 1) of all patterns of occurrence of terms within documents
- Each represents one kind of co-occurrence of index terms in a specific document.

Representations of min-terms

$m_1 = (0, 0, ..., 0)$ ⟺ $\vec{m}_1 = (1, 0, 0, 0, 0, ..., 0)$

$m_2 = (1, 0, ..., 0)$ ⟺ $\vec{m}_2 = (0, 1, 0, 0, 0, ..., 0)$

$m_3 = (0, 1, ..., 0)$ ⟺ $\vec{m}_3 = (0, 0, 1, 0, 0, ..., 0)$

$m_4 = (1, 1, ..., 0)$ ⟺ $\vec{m}_4 = (0, 0, 0, 1, 0, ..., 0)$

$m_5 = (0, 0, 1, ..., 0)$ ⟺ $\vec{m}_5 = (0, 0, 0, 0, 1, ..., 0)$

...

$m_2t = (1, 1, 1, ..., 1)$ ⟺ $\vec{m}_{2^t} = (0, 0, 0, 0, 0, ..., 1)$

2^t minterms — 2^t minterm vectors

Points to the docs where only index terms k_1 and k_2 co-occur and the other index terms disappear.

Pairwise orthogonal vectors $\vec{m}_i$ associated with minterms m_i as the **basis** for the **generalized vector space**.

Point to the docs containing all the index terms.

- Minterm vectors are pairwise orthogonal. But, this does not mean that the index terms are independent.
  - Each minterm specifies a kind of dependence among index terms.
  - That is, the co-occurrence of index terms inside docs in the collection induces dependencies among these index terms.
- The vector associated with the term k_i is represented by summing up all minterms containing it and normalizing.

$$\vec{k_i} = \frac{\sum \forall r\ ,\ g_i\ (m_r) = 1\ c_i,\ r\ \vec{m_r}}{\sqrt{\sum \forall r\ ,\ g_i\ (m_r) = 1\ c_i^2,\ r}} = \sum \forall r,\ g_i\ (m_r) = 1\ \hat{c}_i\ ,\ r\ \bar{m}_r$$

where $$\hat{c}_i\ ,\ r = \frac{c_i,\ r}{\sqrt{\sum \forall r\ ,\ g_i\ (m_r) = 1\ c_i^2,\ r}}$$

$$c_i,\ r = \sum w_{i,\ j}$$

$$\mathbf{d_j \mid g_l\left(\bar{d}_j\right) = g_l\ (m_r),\ for\ all\ l}$$

All the docs whose term co-occurrence relation (pattern) can be represented as (exactly coincide with that of) minterm m_r.

$g_l\ (m_r)$ indicates the index term k_i is in the minterm m_r.

- The weight associated with the pair (k_i, m_r) sums up the weights of the term k_i in all the docs which have a term occurrence pattern given by m_r.
- Notice that for a collection of size N, only N minterms affect the ranking (and not 2^N)

The similarity between the query and doc is calculated in the space of minterm vectors.

$$\vec{d_j} = \sum_i w_{i,j}\, \vec{k_i} \Rightarrow = \sum_r S_{j,r}\, \vec{m_r}$$

$$\vec{q_j} = \sum_i w_{i,q}\, \vec{k_i} \Rightarrow = \sum_r S_{q,r}\, \vec{m_r}$$

t-dimensional 2^t-dimensional

$$\text{sim}\left(\vec{q_j}, \vec{d_j}\right) = \frac{\sum_i w_{i,q} \cdot w_{i,j}}{\sqrt{\sum_i w_{i,q}}\sqrt{\sum_i w_{i,q}}}$$

$$\text{sim}\left(\vec{q_j}, \vec{d_j}\right) = \frac{\sum_r S_{q,r} \cdot S_{d,r}}{\sqrt{\sum_r S_{q,r}}\sqrt{\sum_r S_{d,r}}}$$

Example (a system with three index terms)

minterm	k_1	k_2	k_3
m_1	0	0	0
m_2	1	0	0
m_3	0	1	0
m_4	1	1	0
m_5	0	0	1
m_6	1	0	1
m_7	0	1	1
m_8	1	1	1

$$\vec{k_1} = \frac{C_{1.2}\, \vec{m_2} + C_{1.4}\, \vec{m_4} + C_{1.6}\, \vec{m_6} + C_{1.8}\, \vec{m_8}}{\sqrt{C_{1.2}^2 + C_{1.4}^2 + C_{1.6}^2 + C_{1.8}^2}}$$

$$\vec{k_2} = \frac{C_{2.3}\, \vec{m_3} + C_{2.4}\, \vec{m_4} + C_{2.7}\, \vec{m_7} + C_{2.8}\, \vec{m_8}}{\sqrt{C_{2.3}^2 + C_{2.4}^2 + C_{2.7}^2 + C_{2.8}^2}}$$

$$\vec{k_3} = \frac{C_{3.5}\, \vec{m_5} + C_{3.6}\, \vec{m_6} + C_{3.7}\, \vec{m_7} + C_{3.8}\, \vec{m_8}}{\sqrt{C_{.3.5}^2 + C_{3.6}^2 + C_{3.7}^2 + C_{3.8}^2}}$$

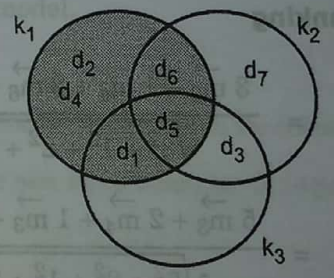


Fig. 2.4.1

$$C_{1.2} = w_{1.2} + w_{1.4} = 1 + 2 = 3$$

$C_{1.4} = w_{1.6} = 1$

$C_{1.6} = w_{1.1} = 2$

$C_{1.8} = w_{1.5} = 1$

$$\vec{k_1} = \frac{3\vec{m_2} + 1\vec{m_4} + 2\vec{m_6} + 1\vec{m_8}}{\sqrt{3^2 + 1^2 + 2^2 + 1^2}}$$

	k_1	k_2	k_3	minterm
d_1	2	0	0	m_6
d_2	1	0	0	m_2
d_3	0	1	3	m_7
d_4	2	0	0	m_2
d_5	1	2	4	m_8
d_6	1	2	0	m_4
d_7	0	5	0	m_3
q	1	2	3	

$C_{2.3} = w_{2.7} = 5$

$C_{2.4} = w_{2.6} = 2$

$C_{2.7} = w_{2.3} = 1$

$C_{2.8} = w_{2.5} = 2$

$$\vec{k_2} = \frac{5\vec{m_3} + 2\vec{m_4} + 1\vec{m_3} + 2\vec{m_8}}{\sqrt{5^2 + 2^2 + 1^2 + 2^2}}$$

$C_{3.5} = 0$

$C_{3.6} = w_{3.1} = 1$

$C_{3.7} = w_{3.3} = 3$

$C_{3.8} = w_{3.5} = 4$

$$\vec{k_3} = \frac{0\vec{m_3} + 1\vec{m_6} + 3\vec{m_7} + 4\vec{m_8}}{\sqrt{0^2 + 1^2 + 3^2 + 4^2}}$$

Example: Ranking

$$\vec{k_1} = \frac{3\vec{m_2} + 1\vec{m_4} + 2\vec{m_6} + 1\vec{m_8}}{\sqrt{3^2 + 1^2 + 2^2 + 1^2}} = \frac{3\vec{m_2} + 1\vec{m_4} + 2\vec{m_6} + 1\vec{m_8}}{\sqrt{15}}$$

$$\vec{k_2} = \frac{5\vec{m_3} + 2\vec{m_4} + 1\vec{m_3} + 2\vec{m_8}}{\sqrt{5^2 + 2^2 + 1^2 + 2^2}} = \frac{5\vec{m_3} + 2\vec{m_4} + 1\vec{m_3} + 2\vec{m_8}}{\sqrt{34}}$$

$$\vec{k_3} = \frac{0\,\vec{m_3} + 1\,\vec{m_6} + 3\,\vec{m_2} + 4\,\vec{m_8}}{\sqrt{0^2 + 1^2 + 3^2 + 4^2}} = \frac{1\,\vec{m_6} + 3\,\vec{m_7} + 4\,\vec{m_8}}{\sqrt{26}}$$

$$\vec{d_1} = 2\,\vec{k_1} + 1\,\vec{k_3}$$

$$= \underset{S_{d1,2}}{\frac{2.3}{\sqrt{15}}}\vec{m_2} + \underset{S_{d1,4}}{\frac{2.1}{\sqrt{15}}}\vec{m_4} + \underset{S_{d1,6}}{\left(\frac{2.2}{\sqrt{15}} + \frac{1.1}{\sqrt{26}}\right)}\vec{m_6} + \underset{S_{d1,7}}{\frac{1.3}{\sqrt{2.6}}}\vec{m_7} + \underset{S_{d1,8}}{\left(\frac{2.1}{\sqrt{15}} + \frac{1.4}{\sqrt{26}}\right)}\vec{m_8}$$

$$\vec{q} = 1\,\vec{k_1} + 2\,\vec{k_2} + 3\,\vec{k_3}$$

$$= \underset{S_{q,2}}{\frac{1.3}{\sqrt{15}}}\vec{m_2} + \underset{S_{q,3}}{\frac{2.5}{\sqrt{34}}}\vec{m_3} + \underset{S_{q,4}}{\left(\frac{1.1}{\sqrt{15}} + \frac{2.2}{\sqrt{34}}\right)}\vec{m_4} + \underset{S_{q,6}}{\left(\frac{1.2}{\sqrt{15}} + \frac{3.1}{\sqrt{26}}\right)}\vec{m_6} + \underset{S_{q,7}}{\left(\frac{2.1}{\sqrt{34}} + \frac{3.3}{\sqrt{26}}\right)}\vec{m_7} + \underset{S_{q,8}}{\left(\frac{1.1}{\sqrt{15}} + \frac{2.2}{\sqrt{34}} + \frac{3.4}{\sqrt{26}}\right)}\vec{m_8}$$

$$\text{sim}(q, d) = \text{consine}(q, d) = \frac{\sum_{r \mid S_{q,r} \neq 0 \wedge S_{d,r} \neq 0} S_{q,*} S_{d,r}}{\sqrt{\sum_{r \mid S_{q,r} \neq 0 \wedge S_{d,r} \neq 0} S_{q,r}^2} \sqrt{\sum_{r \mid S_{q,r} \neq 0 \wedge S_{d,r} \neq 0} S_{q,r}^2}}$$

The similarity between the query and doc is calculated in the space of minterm vectors.

$$\text{sim}(q, d_1) = \frac{S_{q,2} S_{d,2} + S_{q,4} S_{d,4} + S_{q,6} S_{d,6} + S_{q,7} S_{d,7} + S_{q,8} S_{d,8}}{\sqrt{S_{q,.2}^2 + S_{q,.3}^2 + S_{q,.4}^2 + S_{q,.6}^2 + S_{q,.7}^2 + S_{q,.8}^2}\ \sqrt{S_{4,.2}^2 + S_{4,.4}^2 + S_{4,.6}^2 + S_{4,.7}^2 + S_{4,.8}^2}}$$

Term Correlation

The degree of correlation between the terms ki and kj can now be computed as

$$\vec{k_i} \cdot \vec{k_j} = \sum_{\forall r \mid g_i(m_r) = 1 \wedge g_j(m_r) = 1} \hat{C}_{i,r} \times \hat{C}_{j,r}$$

☞ **Advantages of Generalized Vector Model**

1. Model considers correlations among index terms.
2. Model does introduce interesting new ideas.

☞ **Disadvantages of Generalized Vector Model**

1. Not clear in which situations it is superior to the standard vector model.
2. Computation cost is fairly high with large collections.
   - Since the number of "active" minterm might be proportional to the number of documents in the collections.
   - Despite these drawbacks, the generalized vector model does introduce new ideas which are of importance from theoretical point of view.

2.4.2 Latent Semantic Indexing

- Let $M = (M_{ij})$ be a term-document association matrix with t rows and N columns
- Latent semantic indexing decomposes M using Singular Value Decompositions
  - $M = KSD'$
    - K is the matrix of eigenvectors derived from the term-to-term correlation matrix (MM^t).
    - D^t is the matrix of eigenvectors derived from the transpose of the document-to-docume matrix ($M^t M$).
    - S is an r x r diagonal matrix of singular values, where r = min (t, N) is the rank of M.
- Consider now only the **s largest singular values of S**, and their corresponding columns in K and D^t
  - (The remaining singular values of S are deleted)

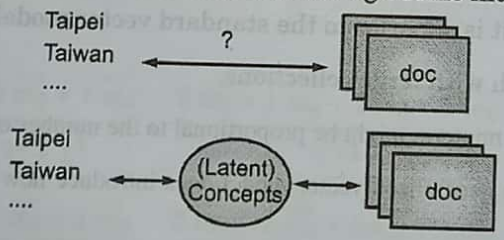
$$Ms = K_s S_s D_s^t$$

- The resultant matrix Ms (rank s) is closest to the original matrix M in the least square sense
- s < r is the dimensionality of a reduced concept space
- The selection of s attempts to balance two opposing effects
  - s should be large enough to allow fitting all the structure in the real data
  - s should be small enough to allow filtering out all the non-relevant representational details
- Consider the relationship between any two documents

$$M^t M = \left(K_s S_s D_s^t\right)^t \left(K_s S_s D_s^t\right) = D_s S_s K_s^t K_s S_s D_s^t = D_s S_s S_s D_s^t$$

$$= (D_s S_s)(D_s S_s)^t$$

- To rank documents with regard to a given user query, we model the query as a pseudo-document in th original matrix M.
  - Assume the query is modelled as the document with number k.
  - Then the k^{th} row in the matrix M'M provides the ranks of all documents with respect to this query
- Latent Semantic Indexing transforms the occurrence matrix into a relation between the terms an concepts, and a relation between the concepts and the documents
  - Indirect relation between terms and documents through some hidden (or latent) concepts.



2.5 PROBABILISTIC INFORMATION RETRIEVAL

GQ. What is probability theory? (2 Marks)
GQ. Explain the working of probabilistic information retrieval model in brief. (6 Marks)
GQ. Explain the Probability Ranking Principle. (4 Marks)
GQ. Explain binary independence model in detail. (6 Marks)

2.5.1 Review of Basic Probability Theory

- A variable A represents an event (a subset of the space of possible outcomes).
- Equivalently, we can represent the subset via a *random variable*, which is a function from outcomes to real numbers; the subset is the domain over which the random variable A has a particular value.
- Often we will not know with certainty whether an event is true in the world. We can ask the probability of the event $0 \leq P(A) \leq 1$. For two events A and B, the joint event of both events occurring is described by the joint probability P (A, B).
- The conditional probability P (A | B) expresses the probability of event A given that event B occurred.

The fundamental relationship between joint and conditional probabilities is given by the *chain rule* :

$$P(A, B) = P(A \cap B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A) \quad \ldots(1)$$

- Without making any assumptions, the probability of a joint event equals the probability of one of the events multiplied by the probability of the other event conditioned on knowing the first event happened.
- Writing $P(\bar{A})$ for the complement of an event, we similarly have :

$$P(\bar{A}, B) = P(B \mid \bar{A})\, P(\bar{A}) \quad \ldots(2)$$

- Probability theory also has a *partition rule*, which says that if an event B can be divided into an exhaustive set of disjoint subcases, then the probability of B is the sum of the probabilities of the subcases. A special case of this rule gives that :

$$P(B) = P(A, B) + P(\bar{A}, B) \quad \ldots(3)$$

- From these we can derive *Bayes´Rule* for inverting conditional probabilities :

$$P(A \mid B) = P(B \mid A)\, P(A) \frac{P(B \mid A)}{P(B)\left[\frac{P(B \mid A)}{\sum_{X \in \{A, \bar{A}\}} P(B \mid X)\, P(X)}\right] P(A)} \quad \ldots(4)$$

- This equation can also be thought of as a way of updating probabilities. We start off with an initial estimate of how likely the event A is when we do not have any other information; this is the *prior probability* P (A).
- Bayes' rule lets us derive a *posterior probability* P (A | B) after having seen the evidence B , based on the *likelihood* of B occurring in the two cases that A does or does not hold.

- Finally, it is often useful to talk about the *odds* of an event, which provide a kind of multiplier for h probabilities change :

$$\text{Odds : } O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1 - P(A)} \quad \ldots(5)$$

2.5.2 Probabilistic Information Retrieval

- As we observed that if we have some known relevant and non-relevant documents, then we can straig forwardly start to estimate the probability of a term t appearing in a relevant document $P(t \mid R = 1$ and that this could be the basis of a classifier that decides whether documents are relevant or not.
- Users start with *information needs*, which they translate into *query representations*.
- Similarly, there are *documents*, which are converted into *document representations* (the latter differin at least by how text is tokenized, but perhaps containing fundamentally less information, as when non-positional index is used).
- Based on these two representations, a system tries to determine how well documents satisf information needs. In the Boolean or vector space models of IR, matching is done in a formally define but semantically imprecise calculus of index terms.
- Given only a query, an IR system has an ambiguous understanding of the information need.
- Given the query and document representations, a system has an uncertain guess of whether document has content relevant to the information need.
- Probability theory provides a principled foundation for such reasoning under uncertainty. It is used t estimate how likely it is that a document is relevant to an information need.

Advantage

Documents are ranked in decreasing order of their probability of being relevant.

Disadvantage

(i) The need to guess the initial relevant and non-relevant sets.

(ii) Term frequency is not considered.

(iii) Independence assumption for index terms.

2.5.3 The Probability Ranking Principle

2.5.3(a) The 1/0 Loss Case

- Using a probabilistic model, the obvious order in which to present documents to the user is to rank documents by their estimated probability of relevance with respect to the information need : $P(R = 1 \mid d, q)$ This is the basis of the *Probability Ranking Principle* (PRP).
- *"If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system.*
- *For this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."*

- In the simplest case of the PRP, there are no retrieval costs or other utility concerns that would differentially weight actions or errors.
- You lose a point for either returning a non-relevant document or failing to return a relevant document (such a binary situation where you are evaluated on your *accuracy* is called 1/0 loss).
- The goal is to return the best possible results as the top k documents, for any value of k the user chooses to examine.
- The PRP then says to simply rank all documents in decreasing order of $P(R = 1 | d, q)$.
- If a set of retrieval results is to be returned, rather than an ordering, the *Bayes Optimal Decision Rule*, the decision which minimizes the risk of loss, is to simply return documents that are more likely relevant than non-relevant :

d is relevant iff $P(R = 1 \mid d, q) > P(R = 0 \mid d, q)$ ...(6)

2.5.3(b) The PRP with Retrieval Costs

- Suppose, instead, that we assume a model of retrieval costs.
- Let C_1 be the cost of not retrieving a relevant document and C_0 the cost of retrieval of a non-relevant document.
- Then the Probability Ranking Principle says that if for a specific document d and for all documents d' not yet retrieved.

$$C_0 \cdot P(R = 0 | d) - C_1 \cdot P(R = 1 | d) \leq C_0 \cdot P(R = 0 | d') - C_1 \cdot P(R = 1 | d') \quad ...(7)$$

then d is the next document to be retrieved. Such a model gives a formal framework where we can model differential costs of false positives and false negatives and even system performance issues at the modeling stage, rather than simply at the evaluation stage.

2.5.4 The Binary Independence Model

- The *Binary Independence Model* (BIM) we present in this section is the model that has traditionally been used with the PRP.
- It introduces some simple assumptions, which make estimating the probability function $P(R|d, q)$ practical.
- Here, "binary" is equivalent to Boolean: documents and queries are both represented as binary term incidence vectors.
- That is, a document d is represented by the vector $\vec{x} = (x_1, \ldots, x_M)$ where $x_t = 1$ if term t is present in document d and $x_t = 0$ if t is not present in d.
- With this representation, many possible documents have the same vector representation.

- Similarly, we represent q by the incidence vector $\vec{q}$ (the distinction between q and $\vec{q}$ is less c since commonly q is in the form of a set of words).
- "Independence" means that terms are modeled as occurring in documents independently. The m recognizes no association between terms.
- This assumption is far from correct, but it nevertheless often gives satisfactory results in practice, the "naive" assumption of Naive Bayes models.
- Indeed, the Binary Independence Model is exactly the same as the multivariate Bernoulli Naive Ba model presented in. In a sense this assumption is equivalent to an assumption of the vector sp model, where each term is a dimension that is orthogonal to all other terms.
- To make a probabilistic retrieval strategy precise, we need to estimate how terms in docume contribute to relevance, specifically, we wish to know how term frequency, document freque document length, and other statistics that we can compute influence judgments about docum relevance, and how they can be reasonably combined to estimate the probability of document relevan We then order documents by decreasing estimated probability of relevance.
- Here we assume here that the relevance of each document is independent of the relevance of oth documents.
- This is incorrect: the assumption is especially harmful in practice if it allows a system to retu duplicate or near duplicate documents.
- Under the BIM, we model the probability P (R|d, q) that a document is relevant via the probability terms of term incidence vectors P (R | $\vec{x}$, $\vec{q}$). Then, using Bayes rule, we have:

$$P(R = 1 \mid \vec{x}, \vec{q}) = -\frac{P\left(\vec{x} \mid R = 1, \vec{q}\right) P\left(R = 1 \mid \vec{q}\right)}{P\left(\vec{x} \mid \vec{q}\right)} \quad \ldots(8$$

$$P(R = 0 \mid \vec{x}, \vec{q}) = -\frac{P\left(\vec{x} \mid R = 0, \vec{q}\right) P\left(R = 0 \mid \vec{q}\right)}{P\left(\vec{x} \mid \vec{q}\right)} \quad \ldots(9$$

- Here, $P(\vec{x} \mid R = 1, \vec{q})$ and $P(\vec{x} \mid R = 0, \vec{q})$ are the probability that if a relevant or non-relevan respectively, document is retrieved, then that document's representation is $\vec{x}$.
- You should think of this quantity as defined with respect to a space of possible documents in a domai How do we compute all these probabilities?
- We never know the exact probabilities, and so we have to use estimates:
- Statistics about the actual document collection are used to estimate these probabilities. $P(R = 1|\vec{q})$ and $P(R = 0 \mid \vec{q})$ indicate the prior probability of retrieving a relevant or non-relevant document respectively for a query $\vec{q}$.

- Again, if we knew the percentage of relevant documents in the collection, then we could use this number to estimate $P(R = 1 | \vec{q})$ and $P(R = 0 | \vec{q})$. Since a document is either relevant or non-relevant to a query, we must have that :

$$P(R = 1 | \vec{x}, \vec{q}) + P(R = 0 | \vec{x}, \vec{q}) = 1 \quad ...(10)$$

2.6 STRUCTURED TEXT RETRIEVAL MODELS

GQ. What is structured text retrieval model? **(4 Marks)**

GQ. What are the models available for Structural Text Retrieval. **(4 Marks)**

- Structured text retrieval models define or provide a mathematical framework for querying semi-structured textual databases.
- A textual database contains content as well as structure.
- The content is the text itself, and the structure divides the database into separate textual parts and connects them using some criterion.
- Textual databases are frequently represented as marked up text, such as XML, where the XML elements define the structure of the text content.
- Textual database retrieval models should be divided into three parts: 1) a text model, 2) a structure model, and 3) a query language:
- The text model specifies tokenization into words or other semantic units, as well as stop words, stemming, synonyms, and so on.
- The structure model defines parts of the text, typically a contiguous portion of the text known as an element, region, or segment, which is defined on top of the text model's word tokens.
- To model relations between content and structure, as well as relationships between structural elements, the query language typically defines a number of operators on content and structure, such as set operators and operators like "containing" and "contained-by."
- Using such a query language, the (expert) user can make requests such as "I want a paragraph discussing formal models near a table discussing the differences between databases and information retrieval."
- The terms "formal models" and "differences between databases and information retrieval" should correspond to the content that needs to be retrieved from the database, whereas "paragraph" and "table" refer to structural constraints on the units to retrieve.
- Retrieval models which combine information on text content with information on the document structure
- That is, the document structure is one additional piece of the information which can be taken advantage
- E.g.: Consider the following information need

- Retrieve all docs which contain a page in which the string 'atomic holocaust' appears in italic in the surrounding a Figure whose label contains the word search.

  Too many doc retrived!

  - ['atomic holocaust' and 'earth']
  - Or a structural (mote complex) query inestead

  data retrieval? Same-page (near ('atomic holocaust, Figure (label ('earth'))))

- **Drawbacks**
  - Difficult to specify the structural query.
  - An advanced user interface is needed.
  - Structured text retrieval models include no ranking (open research problem).
- **Trade-offs**
  - The more expressive the model, the less efficient is its query evaluation strategy.
- **Two structured text retrieval models are –**
  - Non-Overlapping Lists
  - Proximal nodes

☞ **Basic Definitions**

- **Match point :** the position in the text of a sequence of words that match the query.
  - **Query :** "atomic holocaust in Hiroshima"
  - **Doc d_j :** contains 3 lines with this string
  - Then, doc d_j contains 3 match points
- **Region :** a contiguous portion of the text.
- **Node :** a structural component of the text such as a chapter, a section, a subsection, etc.
  - That is, a region with predefined topological properties.

☞ **Structure Text Retrieval**

- Text Retrieval retrieves documents based on index terms.
- **Observation :** Documents have implicit structure.
- Regular text retrieval and indexing strategies lose the information available within the structure.
- Text Retrieval desired based on structure.
- E.g. all documents having "George Bush" in the caption of a photo.

Models for Structural Text Retrieval

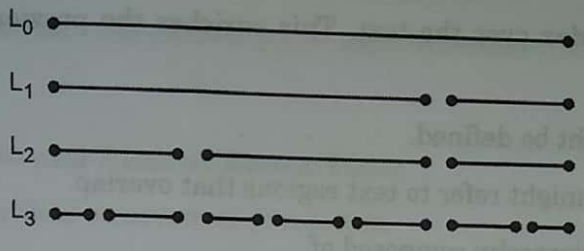
- PAT Expressions
- Overlapped Lists

- Proximal Noded
- Lists of References
- Tree-based
- Query Languages (SFQL,CCL)

Non Overlapping Lists

- **Idea :** divide the whole text of a document in non-overlapping text regions which are collected in a list.
  - Multiple list generated
    - A list for chapters
    - A list for sections
    - A list for subsections

    } 1. Kept as separate and distinct data structures
    2. Text regions from distinct list might overlap



- **Implementation**
  - A single **inverted file** build, in which each structural component stands as an entry in the index.
  - Each entry has a list of text regions as list occurrences.
  - Such a list could be easily merged with the traditional inverted file.
- **Example types of queries**
  - Select a region which contains a given word.
  - Select a region A which does not contain any other region B.
  - Select a region not contained within any other region.

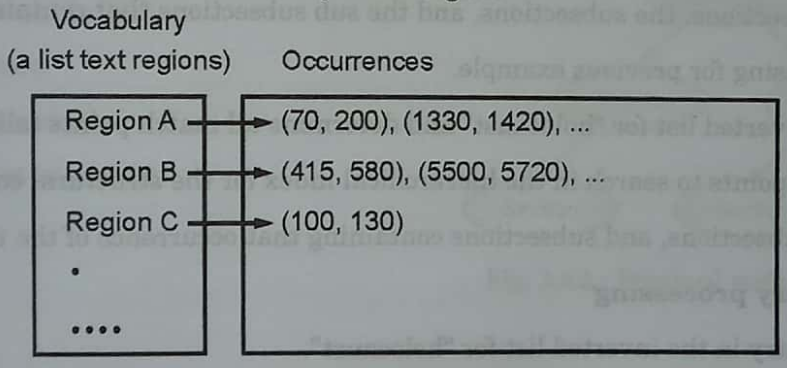


Fig. 2.6.1 : A inverted-file structure for non-overlapping lists

Inverted Files

Definition

- An inverted file is a task-oriented mechanism for indexing a text collection to speed up the search process.

Inverted file structure

1. **Vocabulary :** the collection of all distinct words in the text.
2. **Occurrences :** lists containing all of the information required for each vocabulary word (text position frequency, documents where the word appears, etc.)

Proximal Nodes

- Idea
  - Define a strict hierarchical index over the text. This enriches the previous model that used flat lists.
  - Multiple index hierarchies might be defined.
  - Two distinct index hierarchies might refer to text regions that overlap.
- Each indexing structure is a strict hierarchy composed of
  - Chapters, sections, subsections, paragraphs or lines.
  - Each of these components is called a node.
    - Each node is associated with a text region.
- Query Language in regular expressions
  - Search for strings
  - References to structural components by name
  - Combination of these
- An example query: [(*section) with ("holocaust")]
  - Search for the sections, the subsections, and the sub subsections that contain the word "holocaust".
- Simple query processing for previous example.
  - Traverse the inverted list for "holocaust" and determine all match points (all occurrence entries)
  - Use the match points to search in the hierarchical index for the structural components.
- Look for sections, subsections, and subsections containing that occurrence of the term.
- **Sophisticated query processing**
  - Get the first entry in the inverted list for "holocaust".
  - Use this match pint to search in the hierarchical index for the structural components until innermost matching structural component (the 1st and smallest one) found.
    - At the bottom of the hierarchy.

- Check if innermost matching component includes the second entry in the inverted list for "holocaust".
- If it does, check the two, the third entries... and so on. If not, traverse up higher nodes then traverse down.
- This allows matching efficiently the nearby (or proximal) nodes.

- **Conclusions**
- The model allows formulating queries that are more sophisticated than those allowed by non-overlapping lists.
- To speed up query processing, nearby nodes are inspected.
- Types of queries that can be asked are somewhat limited (all nodes in the answer must come from a same index hierarchy!) **[(*section) with ("holocaust")]**.
- Model is a compromise between efficiency and expressiveness.

Proximal Nodes

- **By Gonzalo Navarro and Ricardo Baeza-Yates**
  - Based on hierarchical structure of documents
  - Structure computation is static and all structural elements are defined. "nodes"
  - Model attempts to define operators on these nodes based on their definition and content.
  - Only nodes at a particular hierarchy are returned as results.
  - Nodes are structural in nature, e.g. Chapter, Section, etc.
- Each node has a defined segment (contiguous part of text)
- Operators are defined with respect to this model.
- Structure operators and Text operators.
- **Structure Operators**
  - Name
  - Inclusion Positional Inclusion
  - Distance operators
  - Child/Parent operators
  - Set Manipulation operators
- **Text Operators**
  - Match

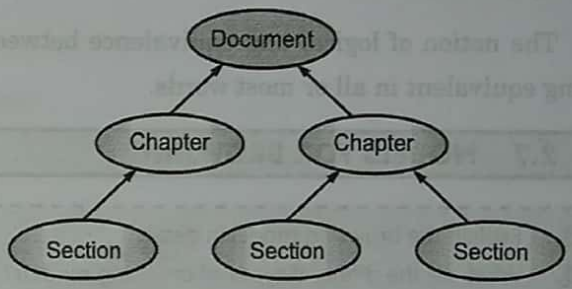


Fig. 2.6.2 : Proximal node representation

Retrieval on Evidence

By Mounia Lalmas

- Based on object-based documents.
- Objects are represented as separate entities that can be in different media, languages, or locations.

- Document indexing - the degree of doubt that the index term truly represents the object.
- To achieve better results, uncertainty must be captured.
- Use the Dempster-Shafer evidence theory.
- Model takes into consideration disparity between indexing vocabularies.
- Aggregation of indexing vocabulary and also the aggregation of the uncertainty.
- Object $o \in O$ and a type $t \in T$, the function type is defined as $O \rightarrow \partial(T)$
- Aggregation is defined over objects and composite object types contain all the types of the conta objects.
- Indexing vocabulary is defined in terms of propositions. Wine (English, text), for example (col feature)
- Sentence space specifies that indexes from the same proposition space can be used in the sa sentence.
- Using the concept of words, the semantic relationship between indexing vocabulary is maintained.
- However, the uncertainty of the representation remains.
- This is represented by the weighting function based on the Dempster Shafer model.
- These objects and their syntactic and semantic models are aggregated for the objects which conta them. E.g. A section containing sentences indexed by terms a, b, c, d.. Will be equivalent to senten over the worlds also implying a,b,c,d...
- Each type t has S, W, V, π
- S_t is the sentence space for a type
- W is the possible words associated with S_t
- v_t is {true, false} over $W_t \times P_t$
- π_t is {true, false} over $W_t \times P_t$

The notion of logical and equivalence between sentences is built around the notion of their semanti being equivalent in all or most words.

2.7 MODELS FOR BROWSING

GQ. Explain the browsing model in detail. **(4 Marks)**

GQ. What are the different types of browsing models? Explain in brief. **(6 Marks)**

- **Premise :** the user is usually interested in browsing the documents instead of searching (specifying th queries)
  - User have goals to purse in both cases.
  - However, the goal of a searching task is clearer in the mind of the user than the goal of a browsin task.

- Three types of browsing are
  - Flat Browsing
  - Structure Guided Browsing
  - The Hypertext Model

Flat Browsing

- Documents represented as dots in
  - A two-dimensional plane
  - A one-dimensional plane (list)
- **Features**
  - Glance here and there looking for information within documents visited
    - Correlations among neighbor document
  - Add keywords of interest into original query
    - Relevance feedback or query expansion
  - Also, explore a single document in a flat manner (like a web page)
- **Drawbacks**
  - No indication about the context where the user is

Structure Guided Browsing

- Documents organized in a **structure as a directory**
  - Directories are hierarchies of classes which **group documents covering related topics**
  - E.g. "Yahoo!" provides hierarchical directory
- Same idea applied to **a single document**
  - Chapter level, section level, etc.
  - The last level is the text itself (flat!)
  - **A good UI needed** for keeping track of the context
  - E.g. the "adobe acrobat pdf" files
- Additional facilities provided when searching
  - A history map identifies classes recently visited
  - Display occurrences (of terms) by showing the structures in a global context, in addition to the text positions

The Hypertext Model

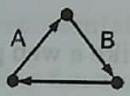
- **Premise :** communication between writer and user
  - A sequenced organizational structure lies underneath most written text

- The reader should not expect to fully understand the message conveyed by the writer by rando reading pieces of text here and there.
- Sometimes, we even can't capture the information through sequential reading of the whole text
  - E.g. : a book about "the history of the wars" is organized chronologically, but we only interes in "the regional wars in Europe"
  - Wars fought by each European country
  - War fought in Europe in chronological order.

  Rewrite the book?

  Or defining a new structure?

- **Hypertext**
  - A high level **interactive navigational structure** allowing users to browse text non-sequentiall
  - Consist of **nodes** (text regions) correlated by directed links in a graph structure
    - A **node** could be a chapter in a book, a section in an article, or a web page
    - Links are attached to specific strings inside the nodes.



- Hypertexts provide the basis for HTML and HTTP
  - HTML : hypertext markup language
  - HTTP : hypertext transfer protocol
- **Features**
  - The process of navigating the hypertext is like a traversal of a directed graph
- **Drawbacks**
  - Loose in hyperspace: the user will lose track of the organizational structure of the hypertext whe it is large.
    - A hypertext map shows where the user is all times (graphical user interface design).
  - But, the user is restricted to the intended flow of information previously convinced by th hypertext designer.
    - Should take into account the need of potential users.
    - Analyzing before the implementation.
    - Guiding tools needed (hypertext map).

Chapter Ends...

❑❑❑