Unit 1

CHAPTER 1

Introduction to Information Retrieval

University Prescribed Syllabus

Basic Concepts of IR, Data Retrieval & Information Retrieval, Text mining and IR relation, IR system block diagram, Automatic Text Analysis: Luhn's ideas, Conflation Algorithm, Indexing and Index Term Weighting, Probabilistic Indexing, Automatic Classification. Measures of Association, Different Matching Coefficients, Cluster Hypothesis, Clustering Techniques: Rocchio's Algorithm, Single pass algorithm, Single Link algorithm.

	Terms in document				
	T1	T2	T3	T4	T5
Doc 1	1	2	0	0	1
Doc 2	3	1	2	3	0
Doc 3	3	0	0	0	1
Doc 4	2	1	0	3	0
Doc 5	2	2	1	5	1

	DI	D2	D3	D4	D5
T1	1	1	0	1	1
T2	2	1	2	3	0
T3	3	0	1	1	1
T4	2	2	0	3	0
T5	2	2	1	2	1

UQ. Why single pass algorithm is better than Rocchio's Algorithm ?

Form the document Clusters of the following documents term matrix using single pass clustering algorithm

Consider

Membership function : sum of product

Centroid calculation function : Average

Threshold = 11. **(SPPU - Q. 2(a), April 17, 10 Marks)** 1-33

	D1	D2	D3	D4	D5
T1	1	1	0	1	1
T2	2	1	2	3	0
T3	3	0	1	0	1
T4	2	2	0	3	0
T5	2	2	1	2	1

UQ. Explain single pass algorithm with example. **(SPPU - Q. 3(a), Dec. 18, 6 Marks)** 1-34

1.14 Single Link algorithm 1-35

UQ. Show how single link clusters may be derived from the dissimilarity coefficient by thresholding it. **(SPPU - Q. 3(b), April 19, 5 Marks)** 1-35

UQ. Dissimilarity matrix is given as follows. **(SPPU - Q. 4(a), May 19, 5 Marks)** 1-35

1

2 0.6

3 0.6 0.8

4 0.9 0.9 0.7

5 0.9 0.6 0.6 0.9

6 0.5 0.5 0.9 0.5 0.5

1 2 3 4 5 6

Threshold 0.4, 0.6, 0.8, 0.9.

Apply single link algorithm and calculate cluster for above 6 objects.

(SPPU - Q. 4(a), May 19, 5 Marks) 1-35

UQ. Explain single link algorithm with example. **(SPPU - Q. 4(a), Dec. 18, 6 Marks)** 1-35

UQ. Explain Single Link algorithm with example. **(SPPU - Q. 2(a), May 2018, 6 Marks)** 1-35

UQ. Show how single link clusters may be derived from the dissimilarity coefficient by thresholding it.

(SPPU - Q. 1(a), May 17, 5 Marks) 1-35

❖ **Chapter Ends...** 1-36

1.1 BASIC CONCEPTS OF IR

1.1.1 Defining Data, Information, And Knowledge, Wisdom

GQ. Define the term: data, information, knowledge, wisdom.

GQ. Justify 'the data is different than the information'.

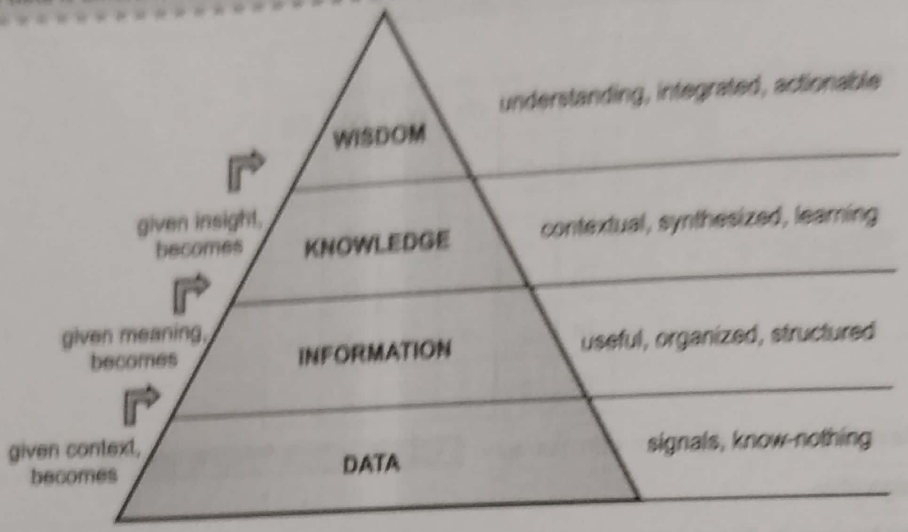


Fig. 1.1.1

- According to Russell Ackoff, a systems theorist and professor of organizational change, the content of the human mind can be classified into five categories :
  (1) **Data :** Symbols
  (2) **Information :** Data that are processed to be useful; provides answers to "who", "what", "where", and "when" questions.
  (3) **Knowledge :** Application of data and information; answers "how" questions.
  (4) **Understanding :** Appreciation of "why"
  (5) **Wisdom :** Evaluated understanding.
- A further elaboration of Ackoff's definitions follows :

(1) Data	(2) Information	(3) Knowledge
(4) Understanding	(5) Wisdom	

▶ **(1) Data**

- This is unprocessed information. It simply exists and has no use (in and of itself). It can exist in any shape or form, whether or not it is usable. It has no intrinsic value. A spreadsheet is a type of computer programme that begins by storing information.
- Data represents a fact or statement of event without relation to other things.

Ex : It is raining.

(2) Information

- Data that has been given meaning through a relationship is referred to as information. This "meaning" may or may not be useful.
- A relational database is a type of database that generates information from the data it holds.
- Information embodies the understanding of a relationship of some sort, possibly cause and effect.

Ex : The temperature dropped 15 degrees and then it started raining.

(3) Knowledge

- Knowledge is a useful collection of data. The process of learning is deterministic. When someone "memorizes" facts (as many less-motivated test-takers do), they have amassed knowledge.
- This knowledge is helpful to them, but it does not provide for an integration that would lead to the acquisition of other knowledge.
- Knowledge represents a pattern that connects and generally provides a high level of predictability as to what is described or what will happen next.

Ex : If the humidity is very high and the temperature drops substantially the atmospheres is often unlikely to be able to hold the moisture so it rains.

(4) Understanding

- Understanding is a probabilistic and interpolative process. It is both cognitive and analytical in nature. It's the method by which I can take previously acquired knowledge and synthesis new information from it.
- The difference between understanding and knowledge is the difference between "learning" and "remembering". People who have understanding can undertake useful actions because they can synthesize new knowledge, or in some cases, at least new information, from what is previously known (and understood).
- That is, understanding can build upon currently held information, knowledge and understanding itself.
- In computer phrasing, AI systems possess understanding in the sense that they are able to synthesize new knowledge from previously stored information and knowledge.

(5) Wisdom

- Wisdom is an extrapolative and non-deterministic, non-probabilistic process. It calls upon all the previous levels of consciousness, and specifically upon special types of human programming (moral, ethical codes, etc.).
- It beckons to give us understanding about which there has previously been no understanding, and in doing so, goes far beyond understanding itself. It is the essence of philosophical probing.
- Unlike the previous four levels, it asks questions to which there is no (easily-achievable) answer, and in some cases, to which there can be no humanly-known answer period. Wisdom is therefore, the process by which we also discern, or judge, between right and wrong, good and bad.

- Wisdom embodies more of an understanding of fundamental principles embodied within the knowledge that are essentially the basis for the knowledge being what it is. Wisdom is essentially systemic.
- **Example :** It rains because it rains. And this encompasses an understanding of all the interactions that happen between raining, evaporation, air currents, temperature gradients, changes, and raining.

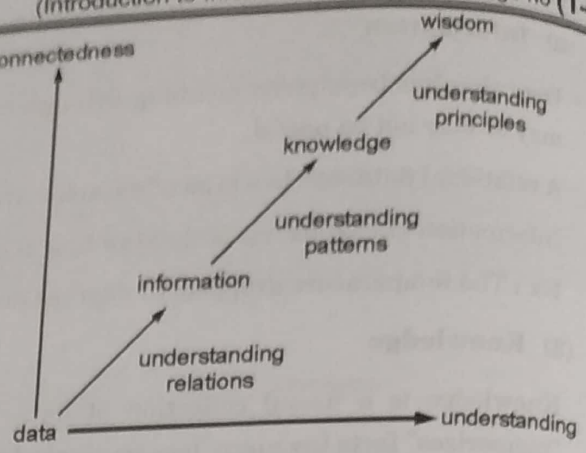
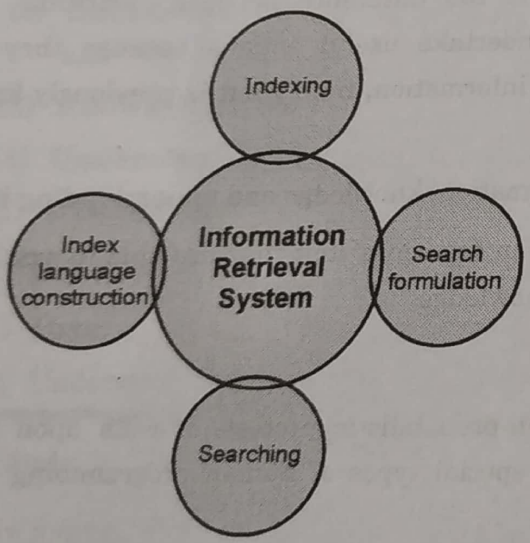


Fig. 1.1.2 : Representation of data, information, knowledge and wisdom

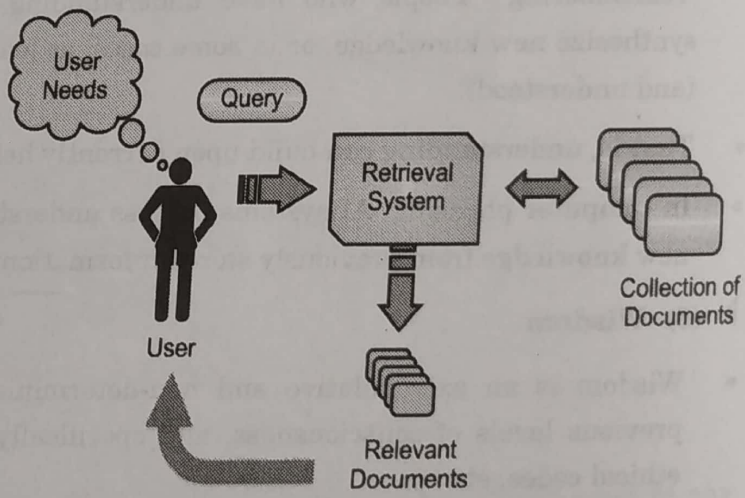
1.1.2 Information Retrieval

GQ. What is information retrieval? **(2 Marks)**
GQ. Define information retrieval. **(2 Marks)**
GQ. How the information retrieval works? **(4 Marks)**
GQ. Explain the importance of information retrieval in brief. **(4 Marks)**
UQ. Draw and explain IR system block diagram. **(SPPU - Q. 1(b), April 19, 5 Marks)**
UQ. Draw and explain IR system block diagram. **(SPPU - Q. 1(a), Dec. 18, 4 Marks)**

- **Information Retrieval** refers to the process, methods, and procedures of searching, locating, and retrieving recorded data and information from a file or database.
- Information retrieval (IR) is the activity of obtaining information system resources that are relevant to an information need from a collection of those resources.



(a) Information Retrieval (IR) System



(b) Information Retrieval (IR) System

Fig. 1.1.3

- *Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds.*

- Modern information retrieval in libraries and archives include searching full-text databases, locating objects from bibliographic databases, and document delivery via a network.
- Automated information retrieval systems are used to reduce what has been called information overload. An Information Retrieval system is a software system that provides access to books, journals, and other documents; stores and manages those documents. Web search engines are the most visible IR applications.
- A print or computer-based system used to search and locate information in a file, database, or other collection of documents is called an information retrieval system. Information retrieval, Recovery of information, especially in a database stored in a computer.
- Two main approaches are matching words in the query against the database index is :
  (1) By using keyword searching
  (2) Traversing the database using hypertext or hypermedia links.
- Keyword searching has been the dominant approach to text retrieval since the early 1960's hypertext has so far been confined largely to personal or corporate information-retrieval applications.
- Natural language, hyperlinks, and keyword searching are all part of evolving information-retrieval approaches, as evidenced by modern Internet search engine improvements.

1.1.3 Data Retrieval and Information Retrieval

GQ. Differentiate between Data retrieval and Information retrieval. **(6 Marks)**

UQ. Write difference between Data retrieval and Information Retrieval. Define Index term. (SPPU - Q. 1(a), Dec. 18, 5 Marks)

UQ. Differentiate between data retrieval and information retrieval. (SPPU - Q. 1(a), May 19, 6 Marks)

UQ. Differentiate between data retrieval and information retrieval. (SPPU - Q. 2(a), Dec. 18, 6 Marks)

Sr. No.	Information Retrieval	Data Retrieval
1.	The software program that deals with the organization, storage, retrieval, and evaluation of information from document repositories particularly textual information.	Data retrieval deals with obtaining data from a database management system such as ODBMS. It is A process of identifying and retrieving the data from the database, based on the query provided by user or application.
2.	Retrieves information about a subject.	Determines the keywords in the user query and retrieves the data.
3.	Small errors are likely to go unnoticed.	A single error object means total failure.
4.	Not always well structured and is semantically ambiguous.	Has a well-defined structure and semantics.
5.	Does not provide a solution to the user of the database system.	Provides solutions to the user of the database system.
6.	The results obtained are approximate matches.	The results obtained are exact matches.
7.	Results are ordered by relevance.	Results are unordered by relevance.
8.	It is a probabilistic model.	It is a deterministic model.

1.2 TEXT MINING AND IR RELATION

GQ. Explain the relationship between IR and text mining. (5 Marks)
UQ. Explain the method for extracting data from text. (SPPU - Q. 10(a), May 16, 8 Marks)

Text Mining

- An vast amount of information and data has resulted from the quick increase of electronic or digital information.
- The majority of the information that is now available is kept in text databases, which are made up of sizable groups of documents from diverse sources. As more information becomes available in electronic form, text databases are expanding quickly.
- More than 80% of the information available today is in the form of unstructured or imperfectly organised data. The rising volume of text data renders obsolete conventional information retrieval methods.
- Thus, text mining has grown in popularity and importance as a component of data mining. In real-world application areas, a significant challenge is finding the right patterns and interpreting the text document from the enormous number of data.

  "Extraction of interesting information or patterns from data in large databases is known as data mining."
- Text mining is the process of removing valuable data and complex patterns from massive text datasets. For the purpose of creating predictions and making decisions, there are numerous methods and tools for text mining. The appropriate and accurate text mining method choice contributes to increased speed and time complexity. In this article, text mining and its uses in various fields are briefly discussed and analysed.

  "Text Mining is the procedure of synthesizing information, by analyzing relations, patterns, and rules among textual data."

The conventional process of text mining as follows :

- Gathering unstructured information from various sources accessible in various document organizations, for example, plain text, web pages, PDF records, etc.
- Pre-processing and data cleansing tasks are performed to distinguish and eliminate inconsistency from the data. The data cleansing process makes sure to capture the genuine text, and it is performed to eliminate stop words stemming (the process of identifying the root of a certain word and indexing the data.
- Processing and controlling tasks are applied to review and further clean the data set.
- Pattern analysis is implemented in Management Information System.
- Information processed in the above steps is utilized to extract important and applicable data for a powerful and convenient decision-making process and trend analysis.

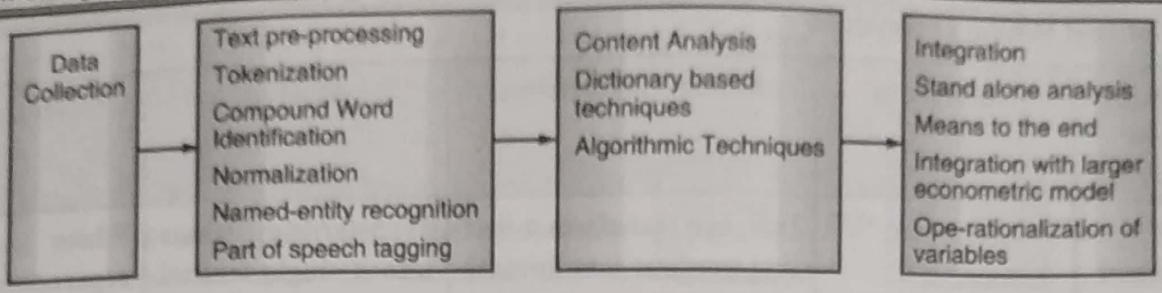


Fig. 1.2.1 : Text mining Process

Procedures of analyzing Text Mining

(1) **Text Summarization :** To extract its partial content reflection its whole content automatically.

(2) **Text Categorization :** To assign a category to the text among categories predefined by users.

(3) **Text Clustering :** To segment texts into several clusters, depending on the substantial relevance.

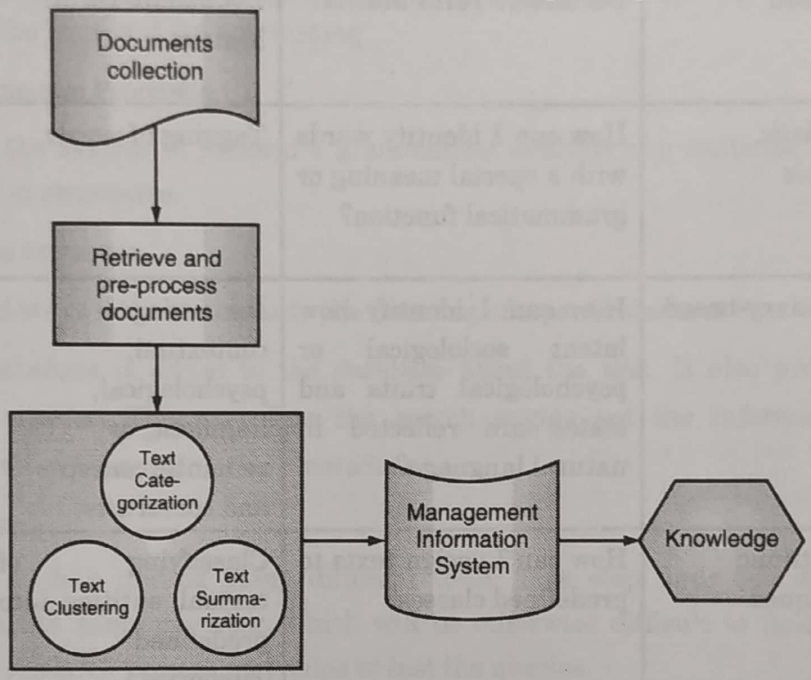


Fig. 1.2.2

TextMining Techniques

(1) **Information Extraction :** It is a process of extract meaningful words from documents.

(2) **Information Retrieval :** It is a process of extracting relevant and associated patterns according to a given set of words or text documents.

(3) **Natural Language Processing :** It concerns the automatic processing and analysis of unstructured text information.

(4) **Clustering :** It is an unsupervised learning process that grouping of text according to their similar characteristics.

(5) **Text Summarization :** To extract its partial content reflection it's whole content automatically.

Overview of Text Mining Techniques

Text Mining Process Phase	Algorithm	Selected Question	Motive	Techniques
1. Text Preprocessing phase	Tokenization	How can transform a text into words or text format?	Transferring strings into a single textual token.	White space separation.
	Compound word identification	How can I identify words that have a joint meaning?	Identifying words with a joint meaning that gets lost word	n-grms
	Normalization and noise reduction	How can I cope with too many variables imy Document-Term-Matrix?	Reducing the dimensionality of Document-Term-Matrix	Stemming, Lemmatization, Deletion of stop words. infrequent term.
	Linguistic analysis	How can I identify words with a special meaning or grammatical function?	Tagging of words	Named-entity recognition, Part-of-speech tagging
2. Content Analysis	Dictionary-based	How can I identify how latent sociological or psychological traits and states are reflected in natural language?	Measuring contextual, psychological, linguistic, or semantic concepts and constructs	pre-defined dictionaries
3. Customized dictionaries	Algorithmic techniques	How can I assign texts to predefined classes?	Classifying of textual entities into predefined categories	Supervised learning techniques such as binary or multi-class classifiers
		How can I group together similar documents?	Clustering of textual entities into formerly undefined and unknown	Unsupervised learning techniques such as LDA, k-means or non-negative

How does Text Mining work?

- Now you should have understood that text mining allows you to understand the text better than anything else. Text Mining system makes an exchange of words from unstructured data into numerical

- Text mining helps to identify patterns and relationships that exist within a large amount of text. Text mining often uses computational algorithms to read and analyze textual information.
- Without text mining, it will be difficult to understand the text easily and quickly. Text can be mined more systematically and comprehensively, and the information about the business can be captured automatically. The steps in the text mining process are listed below.

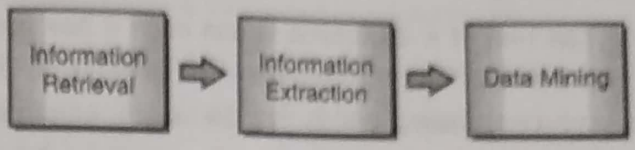


Fig. 1.2.3

▶ **Step 1 : Information Retrieval**

- This is the first step in the process of data mining. This step invol of a sea engine's helpgine to find out the collection of text, also known as a corpus of tethathich might need some conversion.
- These texts should also be brought together in a particular format that will help the users understand. Usually, XML is the standard for text mining.

▶ **Step 2 : Natural Language Processing**

This step allows the system to perform a grammatical analysis of a sentence to read the text. It also analyzes the text in structures.

▶ **Step 3 : Information extraction**

- This is the second stage where to identify the meaning of a particular text mark-up is done.
- In this stage, metadata is added to the database about the text. It also involves adding names or locations to the reader. This step lets the search engine get the information and find out the relationships between them using their metadata.

▶ **Step 4 : Data Mining**

The final stage is data mining using different tools. This step finds the similarities between the information with the same meaning, which will be otherwise difficult to find. Text Mining is a tool which boosts the research process and helps to test the queries.

1.3 IR SYSTEM BLOCK DIAGRAM

1.3.1 Information Retrieval (IR) System

UQ.	Explain basic concept for Information Retrieval. Draw IR system block diagram.	(SPPU - Q. 3(b), May 19, 5 Marks)
GQ.	Draw and explain the information retrieval process?	(4 Marks)
GQ.	Explain the process of information retrieval with suitable example?	(6 Marks)
GQ.	Discuss the classical problem in information retrieval (IR) model?	(4 Marks)
GQ.	Explain how to represent the information retrieval model in mathematical terms?	(4 Marks)
GQ.	Explain with suitable diagram the components of information retrieval in detail.	(6 Marks)
GQ.	Differentiate between the data retrieval information retrieval.	(4 Marks)
GQ.	What are the applications of IR?	(4 Marks)
GQ.	Give the functions of information retrieval system.	(4 Marks)

- Three elements are depicted in the Fig. 1.3.1 input, processor, and output. Putting the input side of things first. Getting a computer-friendly representation of each document and query is the key challenge here.
- The text of a document is lost once it has been processed for the purpose of generating its representation, which is the main emphasis because the majority of computer-based retrieval systems only maintain a representation of the document (or query).
- An example of a document representative would be a list of words that were extracted and deemed important. An alternate method is to create a natural language that all inquiries and documents can be written in, rather than having the machine process the actual language.

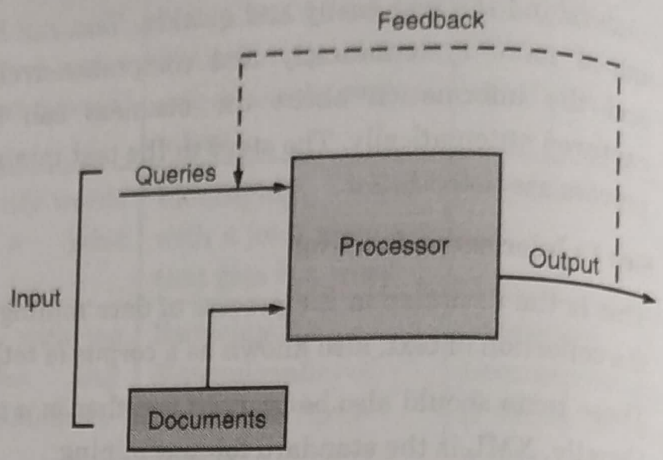


Fig. 1.3.1 : Typical IR systeM

- There is some proof that this can work. Of course, this assumes that a user is open to learning how to communicate his information needs in the language.
- When the retrieval system is online, the user can modify his request during one search session in light of a sample retrieval, hopefully improving the next retrieval run. Feedback is the term used most frequently to describe such a process. The MEDLINE system is a prime illustration of an advanced online retrieval system.
- Second, the retrieval system's processor, which handles the retrieval procedure. The procedure might entail classifying the information or perhaps arranging it in a suitable manner.
- Additionally, it will entail carrying out the retrieval function itself, i.e., carrying out the search strategy in response to a query. The documents have been given their own box in the figure to emphasise that they are not only input but may also be used in a way that the retrieval process can take use of their structure.
- The output, which is typically a list of citations or document numbers, is the last step. In a working system, the narrative finishes here. In an experimental system, however, it leaves the evaluation to be carried out.
- An Information Retrieval system is supported by basic processes such as :

  (1). Query Handling (2) Indexing and (3) Matching
- All these mentioned processes constitutes the Information Retrieval process.
- There are three basic processes an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. The processes are visualized in Fig. 1.3.1. In the figure, squared boxes represent data and rounded boxes represent processes.

- Representing the documents is usually called the indexing process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a formal representation of the document: the index representation or document representation.
- Often, full text retrieval systems use a rather trivial algorithm to derive the index representations, for instance an algorithm that identifies words in an English text and puts them to lower case.
- The indexing process may include the actual storage of the document in the system, but often documents are only stored partly, for instance only title and abstract, plus information about the actual location of the document.
- The query formulation process refers to the process of representing an information problem or need. The query is the resultant formal representation.
- In a broad sense, query formulation may refer to the complete interactive dialogue between system and user that leads not only to a suitable query but also to a better understanding of the user's information need by the user.

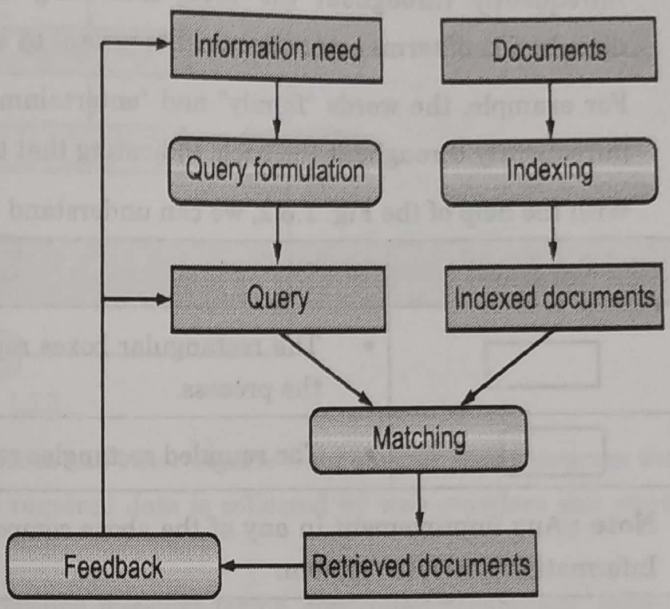
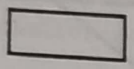
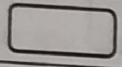


Fig. 1.3.2 : Information Retrieval (IR) Process

- However, in this thesis, query formulation refers to the automatic formulation of the query when no previously retrieved documents are available to guide the search, i.e. the formulation of the initial query.
- The automatic formulation of successive queries is referred to as relevance feedback. The user and the system communicate the information need through queries and retrieved sets of documents, respectively. This is not the most natural way of communicating.
- Humans would use natural language to communicate information needs to one another. A request is a natural language statement of information need. Automatic query formulation takes in the request and generates an initial query.
- In practise, this means that some or all of the words in the request are converted to query terms, such as by the rather simple algorithm that lowercases words. Relevance feedback uses a query or request as input and some previously retrieved relevant and non-relevant documents to generate a subsequent query.
- The matching process refers to the comparison of the query against the document representations.
- The matching process yields a prioritised list of relevant documents. Users will scroll down this document list looking for the information they require.

- Ranked retrieval should place relevant documents near the top of the ranked list, reducing the amount of time the user has to spend reading the documents.
- The frequency distribution of terms across documents is used by simple but effective ranking algorithms.
- For example, the words "family" and "entertainment" mentioned in the first section appear relatively infrequently throughout the book, indicating that this book should not be read. The frequency distribution of terms across documents is used by simple but effective ranking algorithms.
- For example, the words "family" and "entertainment" mentioned in the first section appear relatively infrequently throughout the book, indicating that this book should not be read.
- With the help of the Fig. 1.3.2, we can understand the process of information retrieval (IR) :

Notation	Represents
	• The rectangular boxes represent the information that is supplied as input to the process.
	• The rounded rectangles represent the processes.
Note : Any improvement in any of the above components will result in the improvement of the overall Information Retrieval system.	

- The user's information need is normally referred to as a query. The process of translating the information need into a query is called as query formulation.
- In its original form, a query consists of keywords, and the documents containing those keywords are searched for. A query may consist of a single word or a combination of words with multiple operations.
- Indexing happens in the back end without the direct involvement of the user and is responsible for the representation of the documents.
- The indexing process includes storing the document either partly or in some case the whole document. The index is always built before the searching begins and is a constant dynamic process.
- The query representation is further matched with the document representation that is stored in the index file and is referred to as the **matching process**. It results in a set of ordered documents based on the relevance and is referred to as the **ranked list**.

1.3.2 Components of Information Retrieval / IR Model

GQ. What are the components of IR? (4 Marks)

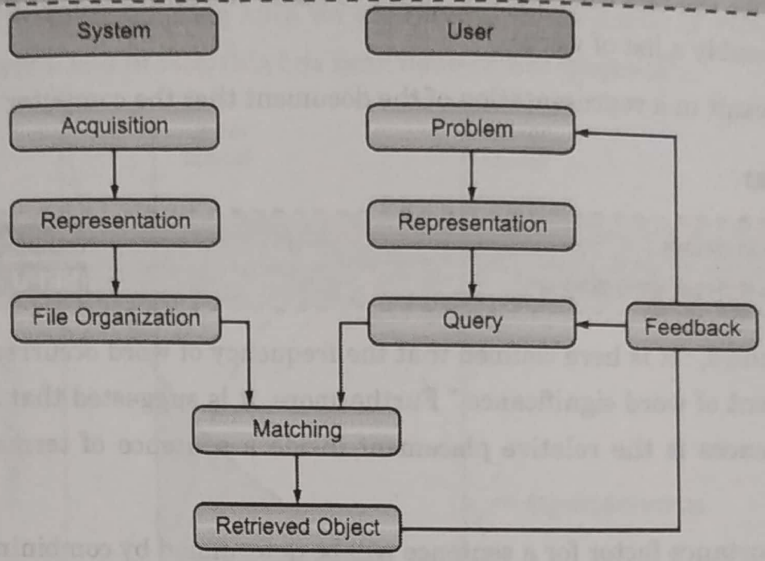


Fig. 1.3.3

(1) **Acquisition :** In this step, the selection of documents and other objects from various web resources that consist of text-based documents takes place. The required data is collected by web crawlers and stored in the database.

(2) **Representation :** It consists of indexing that contains free-text terms, controlled vocabulary, manual & automatic techniques as well.

Example : Abstracting contains summarizing and Bibliographic description that contains author, title, sources, data, and metadata.

(3) **File Organization :** There are two types of file organization methods. i.e. ***Sequential*** **:** It contains documents by document data. ***Inverted*** **:** It contains term by term, list of records under each term. *Combination* of both.

(4) **Query :** An IR process starts when a user enters a query into the system. Queries are formal statements of information needs.

For example : search strings in web search engines. In information retrieval, a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, perhaps with different degrees of relevancy.

1.4 AUTOMATIC TEXT ANALYSIS : LUHN'S IDEAS

GQ. Explain the need of automatic text analysis (4 Marks)

1.4.1 Automatic Text Analysis

- It's unlikely that the computer has saved every document in its entirety in the original natural language, however; rather, it will have a document representation that might have been created manually or automatically from documents.

- Information must be stored inside the computer before a computerised IR system can truly use it to get it.
- The starting point for the text analysis procedure could be the entire text of the document, an abstract, only the title, or possibly a list of words.
- Its creation must result in a representation of the document that the computer can comprehend.

1.4.2 Luhn's Ideas

UQ. Explain Luhn's idea in details. (SPPU - Q. 1(a), May 16, 5 Marks)

UQ. Explain Luhn's idea in detail with diagram. (SPPU - Q. 2(b), Dec. 18, 4 Marks)

- According to the Luhn's, "It is here claimed that the frequency of word occurrence in an article provides a useful measurement of word significance." Furthermore, it is suggested that a helpful indicator of the importance of sentences is the relative placement inside a sentence of terms with different levels of significance.
- As a result, the importance factor for a sentence will be determined by combining these two metrics.
- His assumption is that frequency data can be used to extract words and sentences to represent a document.
- Let f be the frequency of occurrence of various word types in a given position of text and r their rank order, that is, the order of their frequency of occurrence, then a plot relating f and r yields a curve similar to the hyperbolic curve in Fig. 1.4.1.
- This is in fact a curve demonstrating Zipf's Law, which states that the product of the frequency of use of wards and the rank order is approximately constant.
- Zipf verified his law on American Newspaper English. Luhn used it as a null hypothesis to enable him to specify two cut-offs, an upper and a lower (see Fig. 1.4.1), thus excluding non-significant words.
- The terms that were above the upper cut-off were regarded as common, and those that were below the lower cut-off as rare, and as such did not significantly contribute to the article's content.
- As a result, he came up with a counting method for identifying important words. In line with this, he made the assumption that the resolving power of significant words, or the capacity of words to distinguish between content, peaked at a rank order position halfway between the two cut-offs, fell off in either direction from there, and finally decreased to almost zero at the cut-off points.
- Choosing the cut-offs involves some element of arbitrariness. No oracle exists that can reveal their values. They must be developed by trial and error.
- The fact that so many of the later works in IR use these concepts, which are actually quite simple, is intriguing.
- They served as the basis for an automatic abstracting technique developed by Luhn himself. The number of significant and non-significant words in each section of the sentence was used to create a numerical measure of the significance of sentences.

- The highest scoring sentences were included in the abstract after being ranked by their numerical score (extract really).
- There is no justification for limiting such an analysis to words alone. It could also be used to analyse word (or phrase) stems, and in fact, this has been done rather frequently.

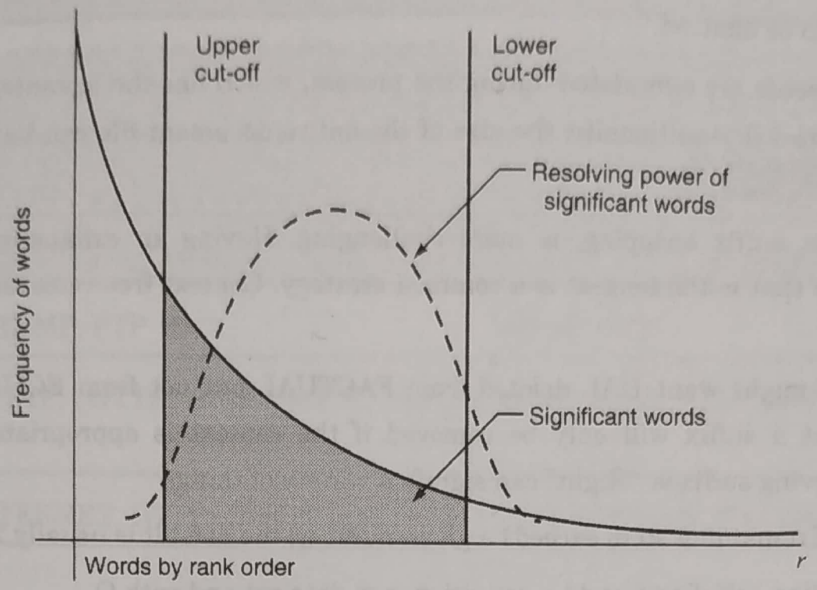


Fig. 1.4.1 : A plot of the hyperbolic curve relating f, the frequency of occurrence and x, the rank outer

1.5 CONFLATION ALGORITHM

1.5.1 Generating Document Representatives – Conflation

GQ. Discuss the idea about generating document representatives –conflation. **(6 Marks)**

GQ. Explain the conflation algorithm. **(6 Marks)**

UQ. Explain working of conflation algorithm in detail justify use of this algorithm in information retrieval? **(SPPU - Q. 1(a), April 17, 6 Marks)**

UQ. Explain steps in conflation algorithm using a suitable example. **(SPPU - Q. 2(a), April 19, 5 Marks)**

UQ. List and explain steps of conflation algorithm. **(SPPU - Q. 2(b), May 19, 4 Marks)**

UQ. You are developing a text processing system for use in an automatic retrieval system. What are different steps of conflation algorithm. **(SPPU - Q. 1(a), Dec. 18, 6 Marks)**

- The objective is to develop a text processing system that can employ computational methods and the least amount of human intervention to extract a document representative from the input text (full text, abstract, or title) that is acceptable for use in an automatic retrieval system.
- This is a challenging request that can only be partially complied with. The document representation I'm aiming for is one that is just a list of class names, with each name representing a set of terms that are present throughout the full input text. A document will be indexed under that name if one of its key phrases is a member of that class.Such a system will usually consist of three parts:

  (1) Removal of high frequency words

(2) Suffix stripping

(3) Detecting equivalent stems.

- Implementing Luhn's upper cut-off involves removing high frequency words, "stop" words, and "fluff" words, among other things. Typically, this is accomplished by comparing the input text to a "stop list" of terms that need to be omitted.
- Non-significant words are eliminated during the process, which has the advantage that they won't get in the way of retrieval. Additionally, the size of the entire document file can be decreased by between 30% and 50%.
- The second stage, suffix stripping, is more challenging. Having an exhaustive list of suffixes and removing the one that is the longest is a common strategy. Context free removal, however, has a high mistake rate.
- For instance, we might want UAL deleted from FACTUAL but not from EQUAL. Context rules are developed so that a suffix will only be removed if the context is appropriate in order to prevent accidentally removing suffixes. "Right" can signify a variety of things :

(1) The length of remaining stem exceeds a given number; the default is usually 2;

(2) The stem-ending satisfies a certain condition, e.g. does not end with Q.

- By dropping their suffixes, several words that are identical in the sense stated above map to a single morphological form. Sadly, while they are equivalent, some others do not.
- This latter type is the one that needs special consideration. Creating a list of equivalent stem-endings is probably the simplest way to handle it. Two stems must be identical aside from their endings, which must also be found in the list as equivalent for them to be considered equivalent.
- For example, stems such as ABSORB- and ABSORPT- are conflated because there is an entry in the list defining B and PT as equivalent stem-endings if the preceding characters match.
- It is assumed (in the context of IR) that if two words share the same underlying stem, they must belong to the same notion and should be indexed as such. Since terms with the same stem, like NEUTRON AND NEUTRALISE, occasionally need to be distinguished, this is plainly a simplification.
- Even words that are essentially equal might have a distinct meaning depending on the situation. Since there is no easy method to make these subtle differences, we accept a certain amount of errors and make the correct assumption that they won't significantly reduce retrieval efficacy.
- Errors will inevitably occur in a processing system like this. Surprisingly, the processing time of this type of algorithm is what limits it, not the size of its core.
- A set of classes, one for each recognized stem, is the algorithm's output in its final form. If and only if one of its constituents appears as a significant word in the document's text, a class name will be given to that piece of writing. The list of class names that follows is once again a document representation. These are also known as the keywords or index terms for the document.

- Naturally, queries are handled in the same way. They may be processed concurrently with the documents in an experimental setting. When a query is submitted to a retrieval system in an operational setting, the text processing system must be used.

1.6 INDEXING AND INDEX TERM WEIGHTING

GQ. Explain the term Indexing. (4 Marks)

UQ. What is term weighting ? Explain the TF IDF scheme to calculate the weight of index term. Find the weight of following terms. (SPPU - Q. 1, May 2018, 10 Marks)

Document	Text	Terms
D1	SNMP, SNMP, FTP	SNMP, FTP
D2	HTTP, FTP, HTTP, ARP, HTTP, SNMP, HTTP	SNMP, FTP, HTTP, ARP
D3	NIC, INTERNET, HTTP, PROTOCOL, HUB	NIC, HTTP, PROTOCOL, HUB, INTERNET

1.6.1 Indexing

- A document's or request's description is made in an index language. The building blocks of the index language are index terms, which can be taken from the description document's text or developed independently.
- The phrase “pre-coordinate” denotes that terms are coordinated at the time of indexing, while "post-coordinate" denotes that terms are coordinated at the time of searching.
- More exactly, a class of documents may be labelled with a logical combination of any index term in pre-coordinate indexing, whereas in post-coordinate indexing the same class would be discovered during search time by combining the classes of documents labelled with the individual index terms.
- Another distinction is whether or not an index language has a regulated vocabulary. The former speaks of a list of acceptable index phrases that an indexer may use, like those employed, for instance, by MEDLARS. Hierarchical relationships between the index terms may also be one of the language controls. Alternately, someone can assert that some words can only be used as adjectives (or qualifiers). The types of syntactic restrictions that can be applied to a language are virtually limitless.
- The index language that results from the conflation process in the previous section can be categorized as post-coordinate, derived, and uncontrolled. The set of all conflation class names constitutes the lexicon of index words at any point in the evolution of the document collection.
- The type of index language that works best for document retrieval is a subject of considerable debate. The suggestions range from complex relational languages to straightforward index terms that text processing algorithms extract.

- The primary topic of discussion is whether or not automatic indexing is superior to manual indexing. There are many complexity levels for each. However, there seems to be growing evidence that manual and automatic indexing do not benefit from adding complexity in the form of rules that are more complicated than index word weighting.
- The implication is that natural language-based, uncontrolled vocabularies can attain retrieval effectiveness on par with vocabulary that include sophisticated controls. This is really encouraging because it makes it easy to automate the simple index language.

1.6.2 Index Term Weighting

GQ. Explain the Index Term Weighting. **(4 Marks)**

- The exhaustivity of indexing and the specificity of the index language have traditionally been regarded as the two most crucial variables affecting an index language's efficiency. The precise definition of these two phrases has generated a lot of discussion.
- Indexing exhaustivity is the total number of topics that are indexed for a given document, while index language specificity is the degree to which topics may be described in detail by the index language. Quantifying these elements is quite challenging. It is possible for human indexers to roughly rank their indexing in terms of increasing exhaustivity or specificity.
- For automatic indexing, it is more difficult to achieve the same. Because of the predicted impact they have on retrieval effectiveness, the concepts of indexing exhaustivity and specificity should be able to be quantified.
- The relationship between high indexing exhaustivity and high recall and low precision has been established. On the other hand, a low amount of exhaustivity causes low recall and high precision.
- Contrarily, high specificity results in high precision and low memory, etc., whereas low specificity leads to high recall. Therefore, it would appear that for a certain user community, there is an ideal amount of indexing exhaustivity and specificity.
- Exhaustivity, for instance, can be thought to be connected to the number of index terms allocated to a given document, and specificity, to the number of documents to which a given term is assigned in a given collection. This ambiguous link between the two elements is significant since it has to do with how the index words in the collection are distributed.
- The hypothesized associations are in line with the previously indicated observed precision and recall trade-off. Changes in the number of index terms per document result in proportional changes in the number of documents per term, and vice versa.
- When we think back to Luhn's original theories, we recall that he proposed that the rank order of an index term's frequency of occurrence would determine its discrimination power, with the middle frequencies having the strongest discriminating power. His approach was suggested for the extraction of important phrases from a document.

- The same frequency counts can, however, be applied to create a weighting system for the various terms in a document. In reality, a popular weighting method is in use that assigns each index phrase a weight that is precisely proportional to how frequently it appears in the document.
- This system initially seems to defy Luhn's theory, which states that the discrimination power decreases as frequency increases. However, referring back to Fig. 1.4.1 (Luhn's), the scheme would be consistent if the upper cut-off is moved to the point where the peak occurs.
- There have been attempts to apply weighting based on how the index terms are spread across the full collection. The index term vocabulary of a document collection frequently has a Zipfian distribution, which means that if we count the number of documents in which each index term occurs and plot them according to rank order, then we obtain the typical hyperbolic shape.
- If there are N documents and an index term occurs in n of them thena weight of log(N/n) + 1 leads to more effective retrieval than if the term were used unweighted. The weighting appears to be giving greater weight to the more specific phrases if indexing specificity is supposed to be inversely related to the number of documents in which an index term occurs.
- It is possible to characterize the difference between the last method of weighting and the prior one by noting that document frequency weighting emphasizes content description while weighting by specificity tries to emphasis terms' capacity to distinguish one document from another.
- To integrate the two weighting techniques by examining both intra- and inter-document frequencies. Actually, their conclusions are just a continuation of Luhn's. They were able to make a number of inferences by taking into account a term's overall frequency of occurrence as well as its distribution among the documents, or how many times it appears in each document. Regardless of how it is distributed, a phrase with a high total frequency of occurrence is not very effective in retrieval.
- In particular, if the distribution is skewed, middle frequency words are most helpful. Although less so than the intermediate frequency terms, rare terms with skewed distributions are likely to be valuable. Except for those with a high overall frequency, very uncommon terms are likewise extremely useful but are listed last. To more precisely assess these conclusions' validity, the experimental evidence supporting them is lacking.
- There is an intriguing method for determining whether an index is "good" or "poor." They believe that a good index term is one that, when applied to a group of documents, makes them as distinct as possible, whereas a bad term makes them more similar.
- This is measured using a term discrimination value, which for a specific term assesses the rise or fall in the average dissimilarity between documents with the removal of that phrase.
- Because of this, a good term is one that, when removed from the collection of documents, causes the average dissimilarity to fall (and thus increase with its addition), whereas a bad term causes an increase with its removal.
- More separation between documents is thought to improve retrieval efficacy while less separation would reduce retrieval effectiveness. Although on the surface this seems acceptable, what is actually needed is for there to be less of a separation between the relevant and irrelevant documents.

- The key findings have been succinctly outlined, and some formal evidence of improved retrieval performance are provided for techniques based on frequency data.
- For example, the inverse document frequency weighting scheme described above, that is assigning a weight proportional to log (N/n) + 1, is shown to be formally more effective than not using these weights.
- Of course, in order to obtain a proof of this nature, some specific presumptions regarding how to gauge effectiveness and how to match documents with inquiries must be established. They also demonstrate the efficacy of a method for conflating low frequency terms, which improves memory, and a method for condensing high frequency terms into phrases, which improves precision.

1.7 PROBABILISTIC INDEXING

GQ. Explain Probabilistic Indexing. **(6 Marks)**

- The difference between the terms word-type and word-token is crucial to the understanding of their model.
- A token instantiates a type, then a particular occurrence at one point in the text of a document (or abstract) will be a word-token.
- Hence 'the frequency of occurrence of word w in a document' means the number of word-tokens occurring in that document corresponding to a unique word-type.
- They consider the difference in the distributional behavior of words as a guide to whether a word should be assigned as an index term.
- They found that function words were closely modeled by a Poisson distribution over all documents whereas specialty words did not follow a Poisson distribution.
- Specifically, if one is looking at the distribution of a function word w over a set of texts then the probability, f(n), that a text will have n occurrences of the function word w is given by

$$f(n) = \frac{e^{-x} x^n}{n!}$$

- In general the parameter x will vary from word to word, and for a given word should be proportional to the length of the text. We also interpret x as the mean number of occurrences of the w in the set of texts.
- What this means is that a word randomly distributed according to a Poisson distribution is not informative about the document in which it occurs.
- At the same time, the fact that a word does *not* follow a Poisson distribution is assumed to indicate that it conveys information as to what a document is about.
- The model also assumes that a document can be about a word to *some degree*.
- This implies that in general a document collection can be broken up into subsets; each subset being made up of documents that are about a given word to the *same degree*.

- The fundamental hypothesis made now is that a content-bearing word is a word that distinguishes more than one class of documents with respect to the extent to which the topic referred to by the word is treated in the documents in each class.
- It is precisely these words that are the candidates for index terms.
- Harter has identified two assumptions, based upon which the above ideas can be used to provide a method of automatic indexing.
- The assumptions are :
  (1) The probability that a document will be found relevant to a request for information on a subject is a function of the relative extent to which the topic is treated in the document.
  (2) The number of tokens in a document is a function of the extent to which the subject referred to by the word is treated in the document.
  (3) The indexing rule based on these assumptions indexes a document with word w if and only if the probability of the document being judged relevant to a request for information on w exceeds some cost function.
  (4) To calculate the required probability of relevance for a content-bearing word we need to postulate what its distribution would look like. We know that it cannot be a single Poisson distribution, and that it is intrinsic to a content-bearing word that it will distinguish between subsets of documents differing in the extent to which they treat the topic specified by the word.
- By assumption (2), within one of these subsets the distribution of a content-bearing can however be described by a Poisson process.
- Therefore, if there are only two such subsets differing in the extent to which they are about a word w then the distribution of w can be described by a mixture of two Poisson distributions.
- Specifically, with the same notation as before we have

$$f(n) = \frac{p_1 e^{-x_1} x_1^n}{n!} + \frac{(1-p_1) e^{-x_2} x_2^n}{n!}$$

here p_1 is the probability of a random document belonging to one of the subsets and x_1 and x_2 are the mean occurrences in the two classes.

- It is important to note that it describes the statistical behavior of a content-bearing word over two classes which are 'about' that word to different extents, these classes are not necessarily the relevant and non-relevant documents although by assumption (1) we can calculate the probability of relevance for any document from one of these classes.
- It is the ratio

$$\frac{p_1 e^{-x_1} x_1^k}{p_1 e^{-x_1} x_1^k + (1-p_1) e^{-x_2} x_2^k}$$

process.

1.8 AUTOMATIC CLASSIFICATION

GQ. Write a short note on: Automatic Classification. (4 Marks)

UQ. What is clustering? Explain the use of clustering in IR. (SPPU - Q. 1(b), May 16, 5 Marks)

Automatic Keyword Classification

- To increase the likelihood of returning pertinent documents, many automatic retrieval systems rely on thesaurus modifications to queries and document representatives.
- There are two ways to construct thesauri :
  (1) Words which are deemed to be about the same topic are linked.
  (2) Words which are deemed to be about related things are linked.
- Manual thesauri are semantically based the automatic thesauri tend to be syntactically & statistical based.
- Process of automatic construction of keyword classes.
- There are two main approaches to the use of keyword classification :
  (1) Replace each keyword in document representation by the name of the class in which it occurs.
  (2) Replace each keyword by all the keywords occurring in the class to which it belongs.
- First way improve the recall & second way will improve the precision.

There are two main areas of application of classification methods in IR :

(1) Keyword clustering

- By grouping keywords into themes that are pertinent to the pages of your website, you can use a process called keyword clustering.
- A single cluster consists of the core topic and multiple connected subtopics that support and reference the core topic. By using keyword clustering, you can target more than one or two keywords each page, increasing the likelihood that your content will be found online.

(2) Document clustering

Document clustering is the process of grouping a set of documents into clusters of similar documents.

(1) Documents within a cluster should be similar.

(2) Documents from different clusters should be dissimilar.

Clustering is the most common form of unsupervised learning.

Unsupervised = there are no labeled or annotated data.

1.9 MEASURES OF ASSOCIATION, DIFFERENT MATCHING COEFFICIENTS

GQ. What do you mean by measures of association? Explain in details. (6 Marks)
UQ. List with definition different measures of association. (SPPU - Q. 1(b), May 19, 4 Marks)

- Several classification methods are based on the binary connection between objects. Based on this link, a classification technique can be used to construct a clustering system.
- "Similarity," "association," and "dissimilarity" are some other terms for the relationship. We'll define dissimilarity mathematically later, so for now let's not worry about it.
- The meanings of the other two terms are nearly identical; nevertheless, we'll save the term "association" for the resemblance between entities that have discrete-state features in common.
- A cluster approach makes it possible to locate such a group structure if it is assumed that it is possible to group items in such a way that an object within a group is more similar to the other members of the group than it is to any other object outside the group. The goal of the similarity metric is to express how similar two items are.
- Five commonly employed measures of association are utilised in information retrieval. Assumes that an object is represented by a list of keywords and that the measuring unit $|.|$ defines the size of the set because documents and requests for information retrieval are frequently represented by term or keyword lists.
- We may readily extrapolate to the case where the keywords have been weighted by simply choosing an acceptable measure. The most basic association metric and different matching co-efficients are :

$|X \cap Y|$ simple matching coefficient

- This indicates how many index terms are shared. The sizes of X and Y are not considered by this coefficient.
- The following coefficients that were applied to document retrieval take into account the data that the sizes of X and Y give.

$2\frac{|X \cap Y|}{|X| + |Y|}$ Dice's coefficient

$\frac{|X \cap Y|}{|X \cup Y|}$ Jaccard's coefficient

$\frac{|X \cap Y|}{|X|^{1/2} \times |Y|^{1/2}}$ Cosine coefficient

$\frac{|X \cap Y|}{\min(|X|, |Y|)}$ Overlap coefficient

These may all be considered to be normalised versions of the simple matching coefficient.

- Dissimilarity function can be transformed into a similarity function by a simple transformation of the form $s = (1 + d)$-1 but the reverse is not always true.

search process.

- If P is the set of objects to be clustered, a pairwise dissimilarity coefficient D is a function from $P \times P$ to the non-negative real numbers. D, in general, satisfies the following conditions :

D1 $D(X, Y) \geq 0$ for all $X, Y \in P$

D2 $D(X, X) = 0$ for all $X \in P$

D3 $D(X, Y) = D(Y, X)$ for all $X \in P$

- In fact, many of the dissimilarity coefficients satisfy the triangle inequality :

D4 $D(X, Y) \leq D(X, Z) + D(Y, Z)$

$$\frac{|X \Delta Y|}{|X| + |Y|}$$

where $(X \Delta Y) = (X \cup Y) - (X \cap Y)$ is the symmetric different of sets X and Y. It is simply related to Dice's coefficient by

$$1 - \frac{2\,|X \cap Y|}{|X| + |Y|} = \frac{|X \Delta Y|}{|X| + |Y|}$$

- Instead of representing each document by a set of keywords, we represent it by a binary string where the absence or presence of the i^{th} keyword is indicated by a zero or one in the i^{th} position respectively. In that case :

$$\frac{\sum x_i (1 - y_i) + \sum y_i (1 - x_i)}{\sum x_i + \sum y_i}$$

- Salton considered document representatives as binary vectors embedded in an n-dimensional Euclidean space, where n is the total number of index terms.

$$\frac{|X \cap Y|}{|X|^{1/2} \times |Y|^{1/2}}$$

- If the space is Euclidean then for $X = (x_1, ..., x_n)$ and $Y = (y_1, ..., y_n)$ we get

$$\frac{(X, Y)}{||X||\,||Y||}$$

- This readily generalises to the case where X and Y are arbitrary real vectors (i.e. weighted keyword lists) in which case we write

$$\frac{\sum_{i=1}^{n} x_i y_i}{\left(\sum_{i=1}^{n} x_1^2\right) 1/2 \left(\sum_{i=1}^{n} y_1^2\right) 1/2}$$

- A probabilistic model has been used by some authors to base measures of association. For two discrete probability distributions $P(x_i)$ and $P(x_j)$ it can be defined as follows :

$$I(x_i, x_j) = \sum_{x_i, x_j} P(x_i, x_j), \log \frac{P(x_i, x_j)}{P(x_i)\,P(x_j)}$$

- Thus let $P_1(1), P_1(0)$ and $P_2(1), P_2(0)$ be the probability distributions associated with class I and II respectively.
- Now on the basis of the difference between them we measure the dissimilarity between I and II by what Jardine and Sibson call the *Information Radius*, which is

$$uP_1(1)\log\frac{P_1(1)}{uP_1(1)+vP_2(1)} + vP_2(1)\log\frac{P_2(1)}{uP_1(1)+vP_2(1)} +$$

$$uP_1(0)\log\frac{P_1(0)}{uP_1(0)+vP_2(0)} + vP_2(0)\log\frac{P_1(01)}{uP_1(0)+vP_2(0)}$$

1.10 CLASSIFICATION

GQ. What do you mean by classification? Explain the various classification methods. (6 Marks)

UQ. What is clustering ? Explain the use of clustering in IR. (SPPU - Q. 2(b), Dec. 18, 5 Marks)

Classification methods

- Objects and the descriptions that go with them make up the data. The items could be papers, words, handwritten characters, or even species.
- Depending on how they are structured, the descriptors go by different names :
  (1) Multi-state attributes (e.g. colour)
  (2) Binary-state (e.g. keywords)
  (3) Numerical (e.g. hardness scale, or weighted keywords)
  (4) Probability distributions.
- In terms of certain broad aspects of the final classificatory system, Sparck Jones has offered a very clear and straightforward breakdown of categorization systems.
  (1) Relation between properties and classes
    (a) Monothetic (b) Polythetic
  (2) Relation between objects and classes
    (a) Exclusive (b) Overlapping
  (3) Relation between classes and classes
    (a) Ordered (b) Unordered
- Consider the example of 8 people (1–8) and 8 attributes in Fig. 1.10.1 to demonstrate the fundamental difference (A-H). A + sign indicates that a piece of property is in your possession. The people 1-4 make comprise a polythetic group, with each member holding three of the four A, B, C, and D traits.
- The remaining 4 people can be divided into two monothetic classes, 5, 6, and 7. If the qualities are of a simple kind, such as binary-state attributes, it is particularly simple to distinguish between monothetic and polythetic features. The definitions become harder to use and, in any event, more arbitrary as the qualities become more complicated.

	A	B	C	D	E	F	G	H
1	+	+	+					
2	+	+		+				
3	+		+	+				
4		+	+	+				
5					+	+	+	
6					+	+	+	
7					+	+		+
8					+	+		+

Fig. 1.10.1 : An illustration of the difference between monothetic and polythetic

- Both theoretically and practically, it is significant to distinguish between overlapping and exclusive relations. Numerous classification techniques are actually data-simplifying techniques.
- In the categorization process, data is removed to make it impossible to tell which objects belong to which class.
- Overlapping classes are permitted in an effort to reduce the quantity of data that is wasted or, to put it another way, to have a classification that is somewhat "closest" to the original data.

1.11 CLUSTER HYPOTHESIS

GQ. What do you mean by Cluster Hypothesis? Explain in detail. **(6 Marks)**

- According to the Cluster Hypothesis, materials that are closely related to one another are frequently relevant to the same queries.
- A fundamental tenet of retrieval systems is that documents pertinent to a request are distinguished from those that are not, that is, that relevant documents are more similar to one another than they are to non-related documents. The following test can be used to determine whether this is true for a collection. Calculate the association between each pair of documents:

  (a) both of which are relevant to a request, and

  (b) one of which is relevant and the other non-relevant.

- The relative distribution of a collection's relevant-relevant (R-R) and relevant-non-relevant (R-N-R) relationships can be determined by adding up a set of requests. For two hypothetical collections, X and Y, we may obtain distributions similar to those in Fig. 1.11.1 by plotting the relative frequency against the degree of relationship.

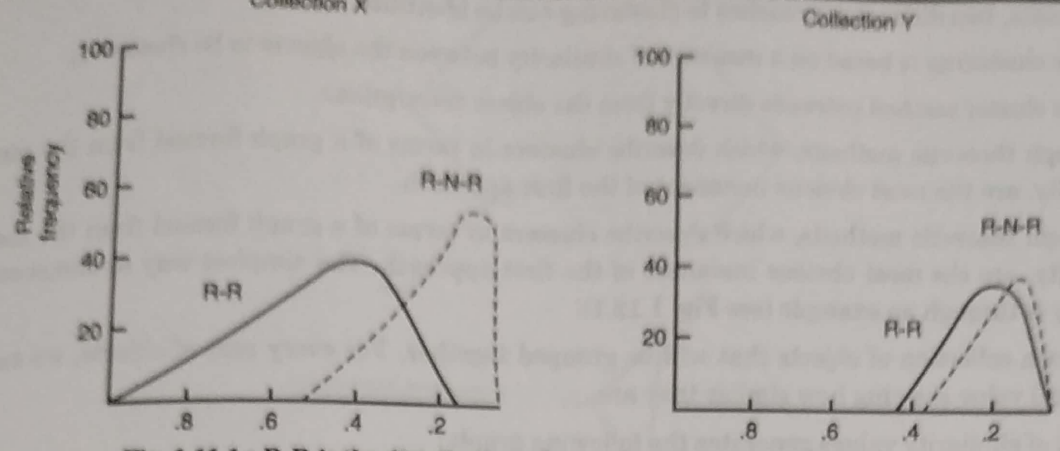


Fig. 1.11.1 : R-R is the distribution of relevant-relevant associations, and R-N-R is the distribution of relevant-non-relevant associations

- The Cluster Hypothesis makes use of specific document descriptions, as should be noted. Thus, the goal of making long- or short-term changes to a description using strategies like keyword classifications can be summed up as an effort to widen the gap between the two distributions R-R and R-N-R.
- In other words, we want to increase the likelihood that we will find relevant papers and decrease the likelihood that we will find irrelevant ones.

1.12 CLUSTERING TECHNIQUES: ROCCHIO'S ALGORITHM

GQ. Explain the use of clustering in information retrieval. **(6 Marks)**

UQ. Explain Riccho's algorithm. **(SPPU - Q. 2(a), Dec. 18, 5 Marks)**

1.12.1 The Use of Clustering in Information Retrieval

In choosing a cluster method for use in experimental IR, two, and criteria have frequently been used.

- The first of these, is the theoretical soundness of the method.
- The method should satisfy certain criteria of adequacy.
- To list some of the more important of these :

(1) The method produces a clustering which is unlikely to be altered drastically when further objects are incorporated, i.e. it is stable under growth.

(2) The method is stable in the sense that small errors in the description of the objects lead to small changes in the clustering;

(3) The method is independent of the initial ordering of the objects.

- The effectiveness of the clustering procedure in terms of speed and storage needs serves as the second factor for selection.
- Efficiency is actually a characteristic of the algorithm used to accomplish the cluster technique.
- It is occasionally helpful to distinguish the cluster method from its algorithm, but in the context of IR this distinction becomes slightly less than useful because many cluster methods are defined by their algorithm, therefore there is no explicit mathematical definition.

relevance judgement
search process

- In the main, two distinct approaches to clustering can be identified :
  (1) The clustering is based on a measure of similarity between the objects to be clustered;
  (2) The cluster method proceeds directly from the object descriptions.
- The graph theoretic methods, which describe clusters in terms of a graph formed from the measure of similarity, are the most obvious instances of the first approach.
- The graph theoretic methods, which describe clusters in terms of a graph formed from the measure of similarity, are the most obvious instances of the first approach. The simplest way to demonstrate this strategy is through an example (see Fig. 1.12.1).
- Consider a collection of objects that will be grouped together. For every pair of objects, we calculate a numerical value showing how similar they are.
- This set of similarity values generates the following graph :

  Two items are deemed related if their similarity values are greater than a predetermined threshold value, which is determined.

  The cluster is easily defined in terms of the graphical display.
- A *string* is a connected series of things starting at one place.
- A *connected component* is a collection of items where each object is linked to at least one other member of the collection and the collection as a whole is *maximal* in terms of this property.
- A *maximum complete subgraph* is one in which every node is connected to every other node in the subgraph and the set is maximal with regard to this property; otherwise, the completeness requirement would be broken if one more node were included anywhere. Each is illustrated in Fig. 1.12.2.

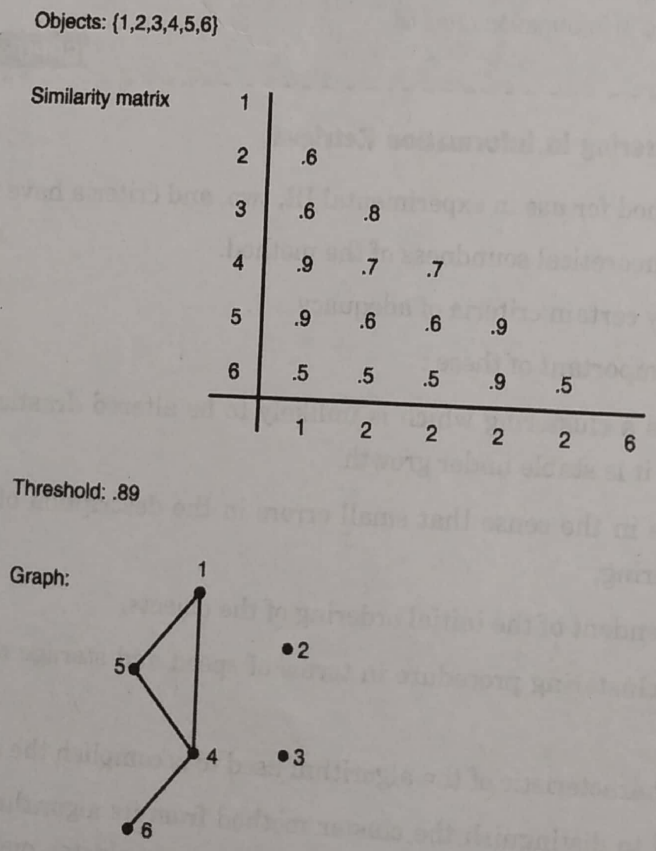


Fig. 1.12.1 : A similarity coefficient for 6 objects and the graph that can be derived from it by thresholding

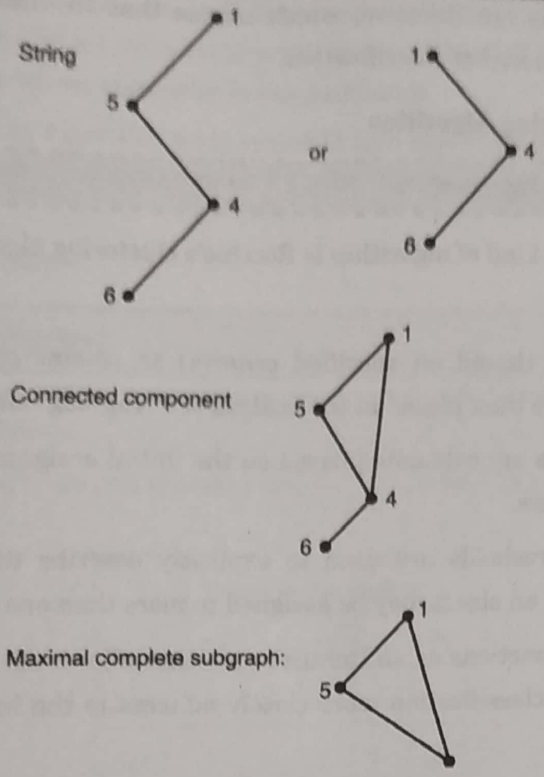


Fig. 1.12.2 : Some possible definitions of clusters in terms of subgraphs

- The most significant of these is single-link, the only one that has been extensively used in document retrieval, and it is based on the initial measurement of similarity that a large class of hierarchic cluster methods are based.
- The so-called "clump" methods are a further class of cluster methods based on the measurement of similarity.
- They work by looking for sets that meet specific cohesion and isolation conditions defined in terms of the similarity measure.
- This strategy has mostly been dropped due to its computational shortcomings.
- An object that summarizes and represents the items in a cluster is referred to as a **cluster representation.**
- Using a matching function, the items' resemblance to the representative is calculated.
- A variety of empirically determined parameters are also used by the algorithms, including :

(1) The intended number of clusters;

(2) The minimum and maximum sizes for each cluster;

(3) The matching function threshold value below which an object will not be included in a cluster;

(4) The management of cluster overlap; and

(5) An arbitrarily selected goal function that is optimised.

relevance judgemen

support for search process.

- The majority of algorithms are iterative, which means that the final classification is produced by incrementally improving an earlier classification.

1.12.2 Rocchio's Clustering Algorithm (6 Marks)

SQ. Explain Rocchio's Clustering Algorithm

The most important of this kind of algorithm is Rocchio's clustering algorithm.

It operates in three stages.

- Several items are chosen (based on specified criteria) as cluster centres in the initial stage. The remainder of the objects are then placed in the centres of a "rag-bag" cluster (for the misfits).
- The cluster representatives are calculated based on the initial assignment, and all items are then once more assigned to the clusters.
- A matching function's thresholds are used to explicitly describe the assignment rules. The latter clusters might overlap (i.e. an object may be assigned to more than one cluster).
- The second stage simply functions as an iterative step that allows the different input parameters to be changed so that the final classification more closely adheres to the initial specification for things like cluster size, etc.
- The third stage is for "tidying up." The amount of cluster overlap is decreased and unassigned items are forcibly assigned.

A traditional Rocchio algorithm, which is a crucial technique for text categorization based vector space models, was introduced by Rocchio in 1971.

The main idea of the algorithm is as follows :

- **Step 1 :** The method divides the initial classes into groups and provides a benchmark data set for comparison. It also labels each sample in the benchmark data set according to the initial classes.
- **Step 2 :** A training set and a test set are created from the data set.
- **Step 3 :** The algorithm selects a class C_i (initially, each class only includes a sample), and W c_i is the representative vector. In the training set, every sample which belongs to C_i, its weight vector's weight expresses by positive, and is added to vector W c_i. To these samples which do not belong to C_i, their weight vector's weight expresses by negative, and is added to vector W c_i. Formula 1 defines the representative vector W c_i :

$$W_{c_i} = \alpha \times \sum_{d_j \in c_i} \frac{\overrightarrow{Wd_j}}{|\overrightarrow{Wd_j}|} - \beta \times \sum_{d_j \notin c_i} \frac{\overrightarrow{Wd_j}}{|\overrightarrow{Wd_j}|} \quad \ldots(1)$$

Two parameters are α and β.

- **Step 4 :** Repeat Step 3 until all classes are selected.

- **Step 5 :** The algorithm selects a document D_i from test set, and the document has not been classified. The similarities between D_i and all class vectors will be computed using the cosine method, and then D_i is labeled to the class Cmax whose similarity is the maximum.
- **Step 6 :** Using Formula 1, the algorithm adds vector D_i into representative vector of the class Cmax, and gets a new vector to represent the class Cmax. To other classes, the vector Di is subtracted from their representative vector.
- **Step 7 :** Repeat Step 5 and Step 6.

1.13 SINGLE PASS ALGORITHM

GQ. Explain the Single pass algorithm. **(6 Marks)**

UQ. Clusters the documents using single pass clustering algorithm for the following example. Threshold value is 10.

(SPPU – Q. 2, May 16, 10 Marks)

	Terms in document				
	T1	T2	T3	T4	T5
Doc 1	1	2	0	0	1
Doc 2	3	1	2	3	0
Doc 3	3	0	0	0	1
Doc 4	2	1	0	3	0
Doc 5	2	2	1	5	1

UQ. Why single pass algorithm is better than Rocchio's Algorithm? Form the document cluster of following document term matrix using single pass clustering algorithm. Consider Membership Function: Sum of product Centroid calculation Function: Average

Threshold = 11

(SPPU – Q. 1, Dec. 16, 10 Marks)

	DI	D2	D3	D4	D5
T1	1	1	0	1	1
T2	2	1	2	3	0
T3	3	0	1	1	1
T4	2	2	0	3	0
T5	2	2	1	2	1

UQ. Why single pass algorithm is better than Rocchio's Algorithm ?
Form the document Clusters of the following documents term matrix using single pass clustering algorithm
Consider
Membership function : sum of product
Centroid calculation function : Average
Threshold= 11.

(SPPU – Q. 2(a), April 17, 10 Marks)

	D1	D2	D3	D4	D5
T1	1	1	0	1	1
T2	2	1	2	3	0
T3	3	0	1	0	1
T4	2	2	0	3	0
T5	2	2	1	2	1

UQ. Explain single pass algorithm with example. **(SPPU – Q. 3(a), Dec. 18, 6 Marks)**

- Only a few clustering techniques require more than one pass over the file of item descriptions. Thus, some of them go by the term "Single-Pass Algorithm." They basically work as follows :
  (1) The object descriptions are processed in serial;
  (2) The first object serves as the first cluster's representative;
  (3) Each succeeding object is compared to all cluster representatives in existence at the time of processing; 4.
  (4) A given object is assigned to one cluster (or more, if overlap is permitted) in accordance with a condition on the matching function.
  (5) The representative for that cluster is recalculated when an object is assigned to that cluster.
  (6) If an object fails a specific test, it becomes the cluster representative of a new cluster.
- The MacQueen algorithm, which begins with an arbitrary initial split of the objects, is comparable to the single-pass strategy. Objects are reallocated to the closest cluster representation when cluster representatives for the partition's members (sets) are computed.
- Like MacQueen, it begins with an initial arbitrary division and a set of cluster representatives. The articles are then redistributed, some of which end up in a cluster known as a "rag-bag." The cluster representative is recomputed after each reallocation, but the new representative won't take the place of the old one unless it turns out that it is anyway closer to the objects in the new cluster.
- The algorithm created by Litofsky should also be mentioned here. His algorithm is only intended to function with objects that have binary state properties. It makes completely distinct use of matching functions and cluster representatives.
- The method moves objects around in an effort to reduce the average amount of unique attributes present in each cluster's members.
- The sets of attribute values that make up the clusters are the sets of attributes shared by all of the cluster members. The final classification is hierarchical.
- The Bonner algorithm should be mentioned lastly. It combines heuristic and graph-theoretic techniques.

- By using graph-theoretical techniques, the initial clusters are defined, and then the items are reallocated based on constraints on the matching function.
- The major advantage of the algorithmically defined cluster methods is their speed order n log n compared with order n^2 for the methods based on association measures. However, they have drawbacks. The cluster algorithm's final classification is dependent on the sequence in which the objects are entered.

1.14 SINGLE LINK ALGORITHM

GQ. Explain the Single Link algorithm. **(6 Marks)**

UQ. Show how single link clusters may be derived from the dissimilarity coefficient by thresholding it.
(SPPU - Q. 3(b), April 19, 5 Marks)

UQ. Dissimilarity matrix is given as follows. (SPPU - Q. 4(a), May 19, 5 Marks)

1					
2	0.6				
3	0.6	0.8			
4	0.9	0.9	0.7		
5	0.9	0.6	0.6	0.9	
6	0.5	0.5	0.9	0.5	0.5
1	2	3	4	5	6

Threshold 0.4, 0.6, 0.8, 0.9.

Apply single link algorithm and calculate cluster for above 6 objects. (SPPU - Q. 4(a), May 19, 5 Marks)

UQ. Explain single link algorithm with example. (SPPU - Q. 4(a), Dec. 18, 6 Marks)

UQ. Explain Single Link algorithm with example. (SPPU - Q. 2(a), May 2018, 6 Marks)

UQ. Show how single link clusters may be derived from the dissimilarity coefficient by thresholding it.
(SPPU - Q. 1(a), May 17, 5 Marks)

- The fundamental input for a single-link clustering technique is the dissimilarity coefficient. The result is a dendrogram, a hierarchy with corresponding numerical levels.
- Often, a tree structure is used to illustrate the hierarchy, with each node standing in for a cluster. Fig. 1.14.1 compares the two representations for the same group of objects (A,B,C,D,E) side by side.
- The clusters are: {A,B}, {C}, {D}, {E} at level L1, {A,B}, {C,D,E} at level L2, and {A,B,C,D,E} at level L3. One can recognise a set of classes at each level of the hierarchy and as one progresses up the hierarchy, the lower level classes are nested inside the higher level classes.

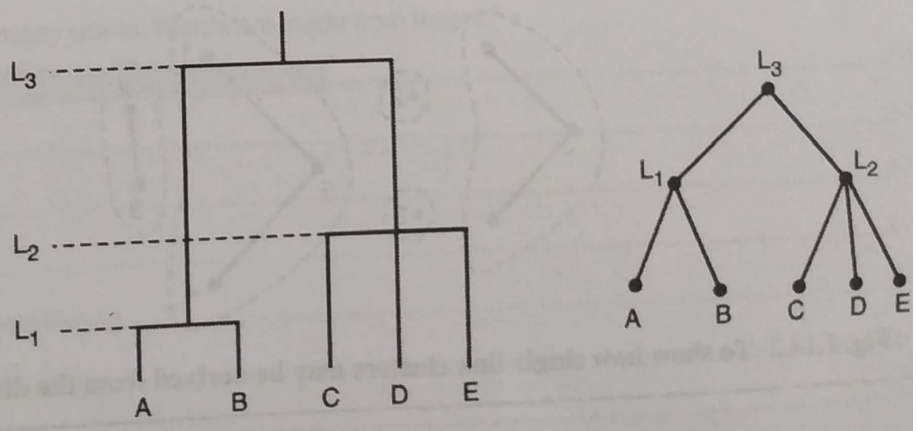


Fig. 1.14.1 : A dendrogram with corresponding tree

- Although there is a mathematical definition of a dendrogram, it is not very useful and will not be included here.
- A worked example is provided to help the reader understand a single-link categorization (see Fig. 1.14.2). A sequence of graphs, one for each value the DC takes, can be used to describe a DC (dissimilarity coefficient). L =.1, 2, 3, and.4 are the various values the DC chose in the example.
- Any two vertices in the graph that correspond to the items to be clustered at a given level are linked if their dissimilarity is at most equal to the level L value.
- Each level is represented by a set of vertices. It should be obvious that these graphs completely describe the DC. A DC can be retrieved from the graphs and their interpretation, and vice versa.
- Compare the graphs at L =.15 and L =.1 as an illustration of how graphs at values other than those taken by the DC are simply the same as at the next smallest value actually taken by the DC.

Dissimilarity matrix:

2	.4			
3	.4	.2		
5	.3	.3	.3	
5	.1	.4	.4	.1
	1	2	3	4

Binary matrices:

2	0			
3	0	0		
5	0	0	0	
5	1	0	0	1
	1	2	3	4

Threshold= .1

2	0			
3	0	1		
5	0	0	0	
5	1	0	0	1
	1	2	3	4

Threshold= .2

2	0		
3	0	1	
5	1	1	1
5	1	0	0
	1	2	3

Threshold= .3

Graphs and clusters:

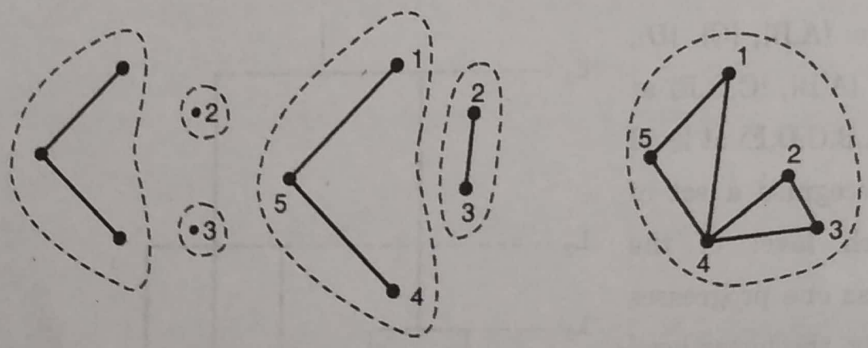


Fig. 1.14.2 :To show how single-link clusters may be derived from the dissimilarity coefficient by thresholding it

Chapter Ends...

□□□

Unit 2

CHAPTER 2

Indexing and Searching Techniques

University Prescribed Syllabus

Indexing : Inverted file, Suffix trees & suffix arrays, Signature Files, Scatter storage or hash addressing.

Searching Techniques : Boolean Search, sequential search, Serial search, cluster-based retrieval, Query languages, Types of queries, Patterns matching, structural queries.

IR Models : Basic concepts, Boolean Model, Vector Model, Probabilistic Model

2.1 INDEXING

GQ. Write a short note on Indexing. **(4 Marks)**

- An essential step in Information Retrieval (IR) systems is indexing. Since it is the first stage in IR and aids in effective information retrieval, it forms the core functioning of the IR process.
- The documents are reduced to their informative terms through indexing. It offers a mapping of the terms to the corresponding documents where they are used. When a collection of documents has an efficient index, retrieval is made simpler.
- The four stages of indexing include content specification, document tokenization, document term processing, and index construction.
- The direct index, document index, lexicon, and inverted index are some of the different data structures in which the index can be kept.
- By utilising various methods or plans, such as single-pass in-memory indexing, blocked-indexing, etc., an index can be created.
- The language used to describe requests and documents is known as an index language.
- The building blocks of the index language are index terms, which can be created either independently or by deriving them from the content of the document to be described.
- Pre-coordinate and post-coordinate index languages are phrases that are coordinated at different times throughout the indexing and searching processes, respectively.
- In more detail, whereas in post-coordinate indexing the same class would be identified at search time by combining the classes of documents labelled with the individual index terms, a logical combination of any index terms may be used as a label to identify a class of documents in pre-coordinate indexing.
- An index is a guide to the items contained in or concepts derived from a collection. Item denotes any book, article, report, abstract review, etc. (textbook, part of a collection, passage in a book, an article in a journal, etc.).
- The word index has its origin in Latin and means: 'to point out, to guide, to direct, to locate'.
- An index indicates or refers to the location of an object or idea. An index is, a working tool designed to help the user to find his way out the mass of documented information in a given subject field, or document store.
- It gives subject access to documents irrespective their physical forms like books, periodical articles, newspapers, AV documents, and computer-readable records including Web resources. (Indexing Process)

2.1.1 Inverted File

2.1.1.1 A Introduction - Inverted File

GQ. What do you mean by inverted files? Explain in detail. (4 Marks)

GQ. Explain the inverted file implementation using sorted array. (6 Marks)

GQ. Discuss the various structures used in inverted files. (6 Marks)

GQ. Explain inverted files concept with suitable example. (6 Marks)

GQ. Discuss the advantages and disadvantages of inverted files. (4 Marks)

UQ. What are inverted files? Explain how these files can be used to answer Boolean queries? (SPPU - Q. 3(b), April 17, 4 Marks)

UQ. Explain inverted File structure with the help of diagram. State how it is useful in implementation of Information Retrieval System. (SPPU - Q. 3(a), May 2018, 5 Marks)

- An inverted file is a data structure that maps content to its location within a database file, a document, or a collection of documents.
- It is typically composed of :
  - a vocabulary containing all of the distinct words found in a text; and
  - a list containing statistics about the occurrences of t in the text for each word t of the vocabulary.
- This type of list is known as the inverted list of t.
- The most common data structure used in document retrieval systems to support full text search is the inverted file.
- The inverted file index concept is as follows. Assume you have a collection of documents.
- Each document is assigned a set of keywords or attributes, with optional relevance weights assigned to each keyword (attribute).
- An inverted file is then a sorted list (or index) of keywords (attributes), with each keyword containing links to the documents that contain that keyword (see Fig. 2.1.1).
- This is the type of index found in most commercial library systems.
- Using an inverted file improves search efficiency by several orders of magnitude, which is essential for very large text files.
- The cost of this efficiency is the need to store a data structure that is 10% to 100% or more the size of the text itself, as well as the need to update that index as the data set changes.

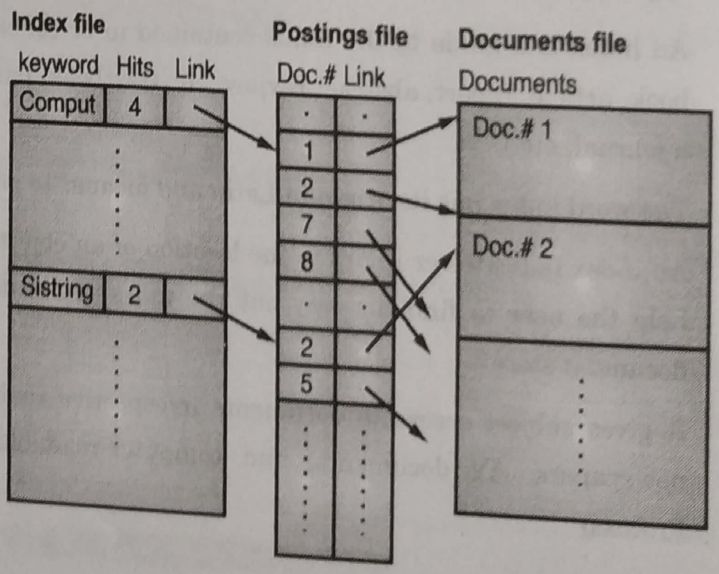


Fig. 2.1.1 : An inverted file implemented using a sorted array

- Typically, some restrictions are imposed on these indices, and thus on subsequent searches. These are some examples of restrictions :
  (1) A controlled vocabulary is a list of keywords that will be indexed. Words in the text that do not appear in the vocabulary will not be indexed and thus will not be searchable.
  (2) A list of stopwords (articles, prepositions, etc.) that will not be included in the index for volume, precision, or recall, and thus are not searchable.
  (3) A set of rules that determines the beginning of a word or a piece of indexable text These rules govern how spaces, punctuation marks, and some standard prefixes are treated, and they may have a significant impact on which terms are indexed.
  (4) A list of indexable character sequences (or not indexed). In large text databases, not all character sequences are indexed; for example, all numeric character sequences are frequently not indexed.
- A search in an inverted file is made up of two searching algorithms : one for a keyword (attribute), which returns an index, and another for a specific attribute value on that index.
- A set of records is the result of a search on an inverted file (or pointers to records).

2.1.2 B Structures Used In Inverted Files

Sorted arrays, B-trees, attempts, and other hashing structures, as well as combinations of these structures, can all be used to implement inverted files.

(1) The Sorted Array (2) B-trees (3) Tries

(1) The Sorted Array

- An inverted file implemented as a sorted array structure stores the list of keywords in a sorted array, including the number of documents associated with each keyword and a link to the documents containing that keyword.
- This array is commonly searched using a standard binary search, though large secondary-storage-based systems will frequently adapt the array (and its search) to the characteristics of their secondary storage.
- The main disadvantage of this approach is that updating the index (for example, adding a new keyword) is costly.
- Sorted arrays, on the other hand, are simple to implement and relatively fast.

(2) B-trees

- A B-tree is another implementation structure for an inverted file.
- The prefix B-tree is a special case of the B-tree that uses word prefixes as primary keys in a B-tree index and is particularly suitable for storing textual indices. Each internal node has a unique set of keys.
- Each key is the shortest (in length) word that distinguishes the keys in the next level.

- The key does not have to be a prefix to a term in the index.
- The final level, or leaf level, stores the keywords and their associated data (see Fig. 2.1.2).
- The order (size) of each node in the prefix B-tree is variable because the internal node keys and their lengths are determined by the set of keywords.
- To maintain a balanced tree, updates are performed similarly to those performed on a B-tree.
- When there are many words with the same (long) prefix, the prefix B-tree method fails.
- To avoid wasting space, common prefixes should be further divided in this case.

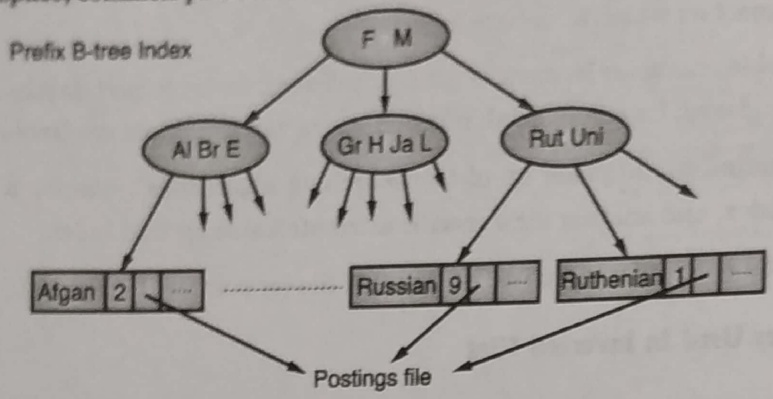


Fig. 2.1.2 : A prefix B-tree

- B-trees consume more space than sorted arrays.
- However, updates are much easier and search times are generally faster, especially if secondary storage is used for the inverted file (instead of memory).
- The implementation of inverted files using B-trees is more complicated than using sorted arrays.

(3) Tries

- Trie is an efficient information retrieval data structure.
- Using Trie, search complexities can be brought to optimal limit (key length).
- If we store keys in a binary search tree, a well-balanced BST will need time proportional to M * log N, where M is the maximum string length and N is the number of keys in the tree.
- An inverted index is a data structure that stores a mapping between content, such as words or numbers, and their locations in a document or set of documents.
- It is a hashmap-like data structure that guides you from a word to a document or a web page.
- Inverted indexes are classified into two types: A record-level inverted index includes a list of document references for each word. A word-level inverted index also includes each word's position within a document.
- The latter provides more functionality, but requires more processing power and storage space to be created.
- Suppose we want to search the texts "hello everyone," "this article is based on inverted index, " "which is hashmap like data structure".

- If we index by (text, word within the text), the index with location in text is:

hello	(1, 1)
everyone	(1, 2)
this	(2, 1)
article	(2, 2)
is	(2, 3); (3, 2)
based	(2, 4)
on	(2, 5)
inverted	(2, 6)
index	(2, 7)
which	(3, 1)
hashmap	(3, 3)
like	(3, 4)
data	(3, 5)
structure	(3, 6)

- The word "hello" is in document 1 ("hello everyone") starting at word 1, so has an entry (1, 1) and word "is" is in document 2 and 3 at '3rd' and '2nd' positions respectively (here position is based on word). The index may have weights, frequencies, or other indicators.

☞ **Steps to build an inverted index**

Fetch the Document

Removing of Stop Words: Stop words are most occurring and useless words in document like "I", "the", "we", "is", "an".

Stemming of Root Word

- Whenever I want to search for "cat", I want to see a document that has information about it. But the word present in the document is called "cats" or "catty" instead of "cat".
- To relate the both words, I'll chop some part of each and every word I read so that I could get the "root word". There are standard tools for performing this like "Porter's Stemmer".

Record Document IDs

If word is already present add reference of document to index else create new entry. Add additional information like frequency of word, location of word etc.

☞ **Example**

Words	Document
ant	doc1
demo	doc2
world	doc1, doc2

user relevance judg

Advantage of Inverted Index are

(1) Inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database.

(2) It is easy to develop.

(3) It is the most popular data structure used in document retrieval systems, used on a large scale for example in search engines.

Inverted Index also has disadvantage

(1) Large storage overhead and high maintenance costs on update, delete and insert.

(2) **Inverted index :** a word-oriented mechanism for indexing a text collection to speed up the searching task.

(3) The inverted index structure is composed of two elements: the **vocabulary** and the **occurrences**.

(4) The vocabulary is the set of all different words in the text

(5) For each word in the vocabulary the index stores the documents which contain that word (inverted index).

(6) Term-document matrix: the simplest way to represent the documents that contain each word of the vocabulary.

Vocabulary	n_i	d_1	d_2	d_3	d_4
to	2	4	2	-	-
do	3	2	-	3	3
is	1	2	-	-	-
be	4	2	2	2	2
or	1	-	1	-	-
not	1	-	1	-	-
I	2	-	2	2	-
am	2	-	2	1	-
what	1	-	1	-	-
think	1	-	-	1	-
therefore	1	-	-	1	-
da	1	-	-	-	3
let	1	-	-	-	2
it	1	-	-	-	2

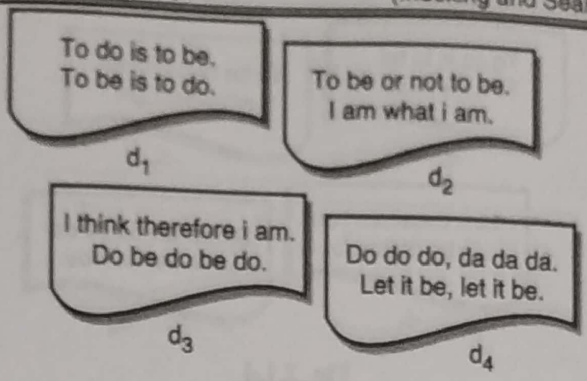


Fig. 2.1.3

- The main problem of this simple solution is that it requires too much space.
- As this is a sparse matrix, the solution is to associate a list of documents with each word.
- The set of all those lists is called the occurrences.

☞ **Basic inverted index**

Vocabulary	n_i	Occurrences as inverted lists
to	2	[1, 4], [2, 2]
do	3	[1, 2] [3, 3], [4,3]
Is	1	[1, 2]
be	4	[1, 2] [2, 2], [3, 2], [4, 2]
or	1	[2, 1]
not	1	[2, 1]
I	2	[2, 2], [3, 2]
am	2	[2, 2], [3, 1]
what	1	[2, 1]
think	1	[3, 1]
therefore	1	[3, 1]
da	1	[4, 3]
let	1	[4, 2]
It	1	[4, 2]

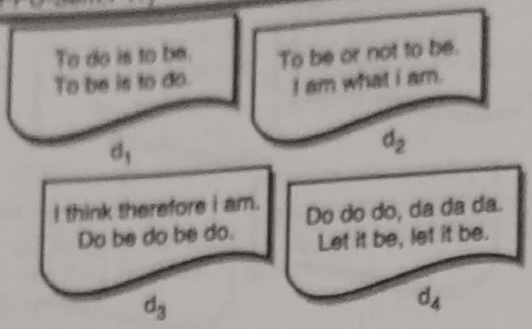
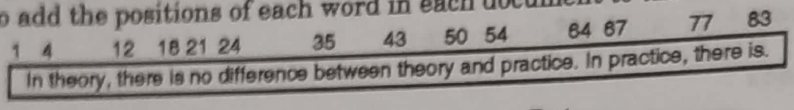


Fig. 2.1.4

Full Inverted Indexes

- The basic index is not suitable for answering phrase or proximity queries.
- Hence, we need to add the positions of each word in each document to the index (full inverted index).



Text

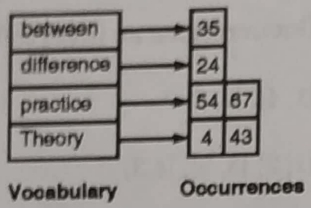


Fig. 2.1.5

- In the case of multiple documents, we need to store one occurrence list per term-document pair.

Vocabulary	n_i	Occurrences as full inverted lists
To	2	[1,4,[1,4,6,9]], [2,2,[1,5]]
Do	3	[1, 2,[2, 10]],[3,3,[6,8,10]],[4,3,[1,2,3]]
Is	1	[1,2,[3,8]]
Be	4	[1,2,[5,7]],[2,2,[2,6]],[3,2,[7,9]],[4,2,[9,12]]
Or	1	[2,1,[3]]
Not	1	[2,1,[4]]
I	2	[2,2,[7,10]],[3,2,[1,4]]
Am	2	[2,2,[8,11]],[3,1,[5]]
What	1	[2,1,[9]]
Think	1	[3,1,[2]]
therefore	1	[3,1,[3]]
Da	1	[4,3,[4,5,6]]
Let	1	[4,2,[7,10]]
It	1	[4,2,[8,11]]

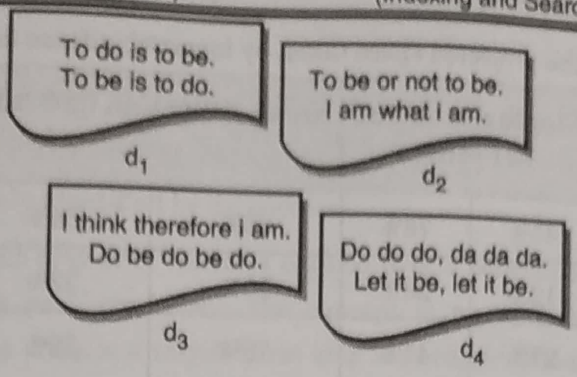
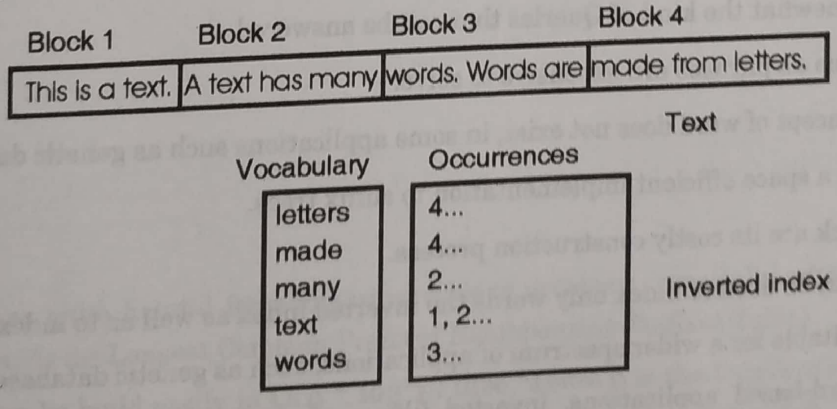


Fig. 2.1.6

- The space required for the vocabulary is rather small.
- Heaps' law: the vocabulary grows as O(nβ),

  Where,
  - n is the collection size.
  - β is a collection-dependent constant between 0.4 and 0.6.
- For instance, in the TREC-3 collection, the vocabulary of 1 gigabyte of text occupies only 5 megabytes.
- This may be further reduced by stemming and other normalization techniques.
- The occurrences demand much more space.
- The extra space will be O(n) and is around.
  - 40% of the text size if stopwords are omitted
  - 80% when stopwords are indexed.
- Document-addressing indexes are smaller, because only one occurrence per file must be recorded, for a given word.
- Depending on the document (file) size, document-addressing indexes typically require 20% to 40% of the text size.
- To reduce space requirements, a technique called block addressing is used.
- The documents are divided into blocks, and the occurrences point to the blocks where the word appears.



- The Table below presents the projected space taken by inverted indexes for texts of different sizes

Index granularity	Single document (1 MB)		Small collection (200 MB)		Medium collection (2 GB)	
Addressing words	45%	73%	36%	64%	35%	63%
Addressing documents	19%	26%	18%	32%	26%	47%
Addressing 64 K blocks	27%	41%	18%	32%	5%	9%
Addressing 256 blocks	18%	25%	1.7%	2.4%	0.5%	0.7%

- The blocks can be of fixed size or they can be defined using the division of the text collection into documents.
- The division into blocks of fixed size improves efficiency at retrieval time.

  This is because larger blocks match queries more frequently and are more expensive to traverse.
- This technique also profits from locality of reference.

  That is, the same word will be used many times in the same context and all the references to that word will be collapsed in just one reference.

2.1.3 Suffix Trees and Suffix Arrays

GQ. Explain the concepts suffix trees and suffix arrays. (6 Marks)

UQ. Give the difference between suffix array and suffix tree (SPPU - Q. 3(b), May 16, 5 Marks)

UQ. Explain working of suffix tree. Construct suffix tree for following example.
"This is a text. A text has many words. Words are made from letters."
(SPPU - Q. 2(a), Dec. 16, Q. 4(a), April 17, 6 Marks)

2.1.3.1 Suffix Trees

- Inverted indices assume that the text can be seen as a sequence of word.
- This restricts somewhat the kind of Queries that can be answered.
- Other queries such as phrases are expensive to solve.
- Moreover, the concept of word does not exist, in some applications such as genetic databases.
- Suffix arrays are a space efficient implementation to suffix trees.
- Its main drawback are its costly construction process.
- This structure can be used to index only words the inverted index as well as to index any text character.
- This makes it suitable for a wider spectrum of applications, such as genetic databases.
- However for word-based applications, inverted files perform better unless complex queries are an important issue.

- This index sees the text as one long string.
- Each position the text is considered as a text suffix.
- It is not difficult to see that two suffixes starting at different positions are lexicographically different.
- Each suffix is thus uniquely identified by its position.

T is a logical advancement over trie, which is defined over a collection of substrings of a string s and utilised in pattern matching problems. Here, the concept is pretty basic. Such a triangle may have numerous long, straight paths. This is a terrific first step toward increasing the difficulty of operations on such a tree since if we can just condense these lengthy paths into a single jump, we will dramatically shrink the trie's size. A suffix tree of s is a reduced trie defined over a subset of a string's suffixes.

For better understanding, let's consider the suffix tree **T** for a string **s = abakan**. A word abakan has 6 suffixes **{abakan , bakan, akan, kan, an, n}** and its suffix tree looks like this:

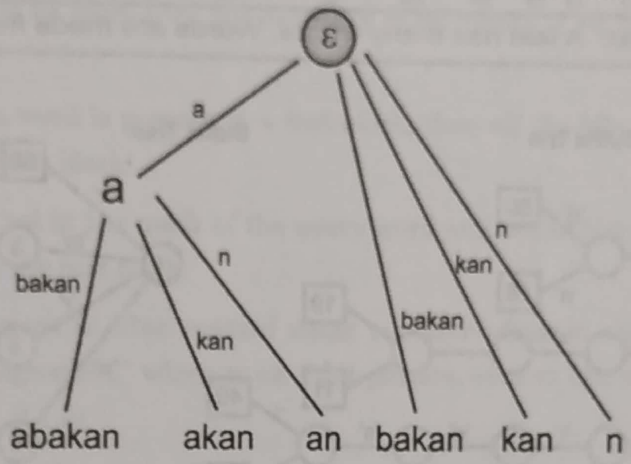


Fig. 2.1.7

2.1.3.2 Suffix Array

- Suffix array is a very nice array based structure. Basically, it is a lexicographically sorted array of suffixes of a string s. For example, let's consider a string s = abakan.
- A word abakan has 6 suffixes {abakan , bakan, akan, kan, an, n} and its suffix tree looks like this:

  0 : Abakan

  1 : akan

  2 : an

  3 : bakan

  4 : kan

  5 : n
- Suffix arrays are quite helpful for addressing various problems, especially when used with the LCP table (which stands for Longest Common Prefix of Neighbouring Suffixes Table).
- Suffix arrays can be build easily in $\mathbf{O(n * \log^2 n)}$ time, where **n** is the length of **s**, using the algorithm proposed in the paper from the previous paragraph. This time can be improved to $\mathbf{O(n * \log n)}$ using linear time sorting algorithm.

- Fig. 2.1.8 (a & b) shows the representation of suffixes, suffix trees and suffix arrays.

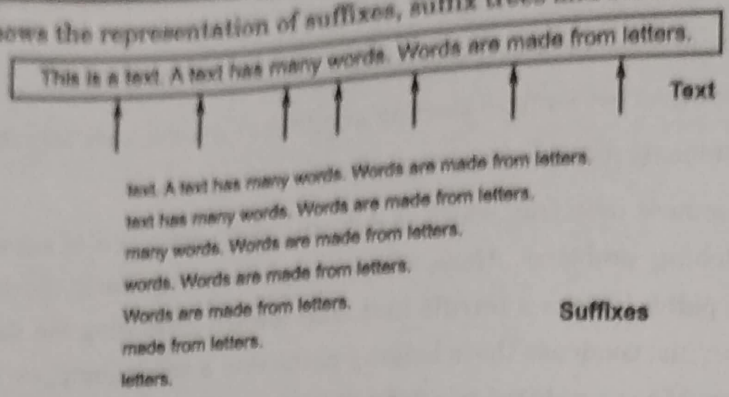


Fig. 2.1.8

(a) Suffixes

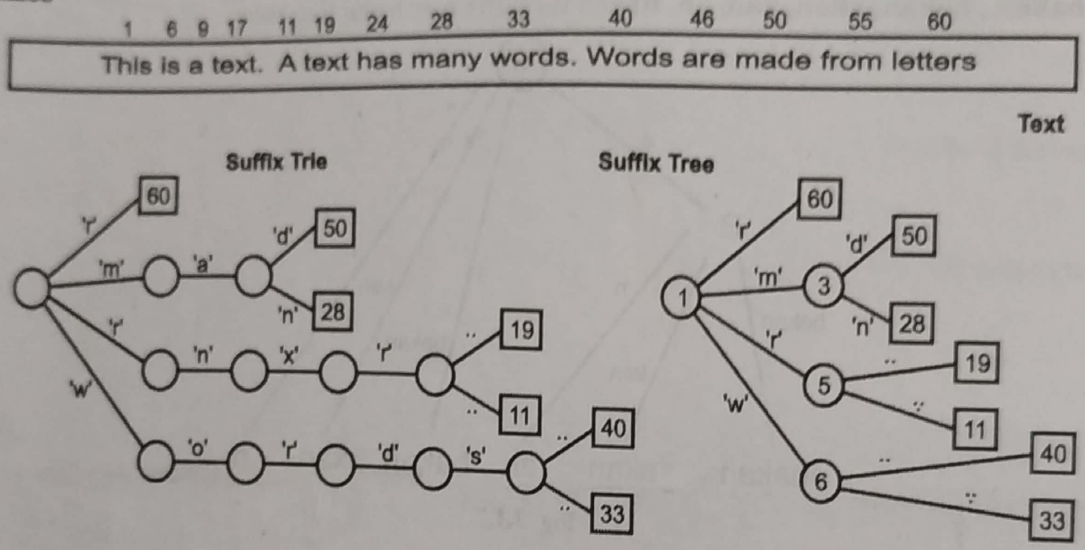


Fig. 2.1.8(a)

(b) Suffix tree and array

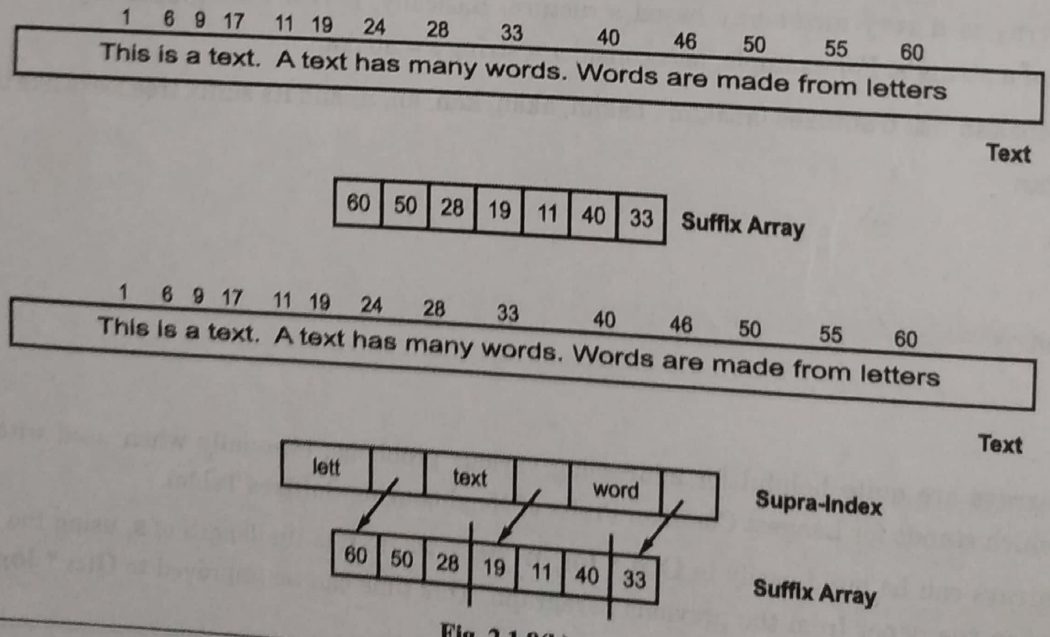


Fig. 2.1.8(b)

2.1.4 Signature Files

GQ. Explain the Signature Files. (4 Marks)

- Signature files are word-oriented index structures based on hashing.
- They pose a low overhead, at the cost of forcing n sequential search over the index.

Structure

- A signature file uses h hash function (or 'signature') that maps words to bit masks of B bits.
- It divides the test in blocks of b words each.
- To each text block of *size* b a hit mask of size B will be assigned.
- This mask is obtained by bitwise ORing the signatures of all the words in the test block.
- Hence, the signature file is no more than the sequence of bit masks of all blocks (plus a pointer to each block).
- The main idea is that if a word is present in a text block, then all the bits set in its signature are also set in the bit mask of the text block.
- Hence, whenever h bit is set in the mask of the query word and not in the mask of the test block, then the word is not present in the text block.
- The signature of a document is often created using overlay coding in signature files. Each of these words produces a "word signature," which is an F-bit pattern with m bits set to "1" and the remaining bits all set to "0." (see Fig. 2.1.9).
- The design parameters are F and m. The block signature is created by ORing the word signatures together. The document signature is created by concatenating block signatures.
- Hash functions choose which m bit places will be set to "1" by each word. The process of finding a word involves producing the word's signature and checking each block signature for "1"s in the bit places where the search word's signature has a "1".

Word	Signature
free	001 000 110 010
text	000 010 101 001
block signature	001 010 111 011

Fig. 2.1.9 : Illustration of the superimposed coding method

- It is assumed that each logical block consists of D = 2 words only. The signature size F is 12 bits, m=4 bits per word.

- The following approach has been suggested to enable word fragment searches.
- Every syllable is broken up into three consecutive, overlapping triplets (e.g., "fr", "fre", "ree", "ee" for the word "free"). By using a hashing algorithm on a triplet's numerical encoding, such as treating the triplet as a base-26 integer, each of these triplets is hashed to a bit position.
- The word is permitted to set l (nondistinct) bits if it contains l triplets and l > m. If l m, a random number generator initialised with a word's numerical encoding is used to set the extra bits.
- Fig. 2.1.10 shows an example of signature creation.

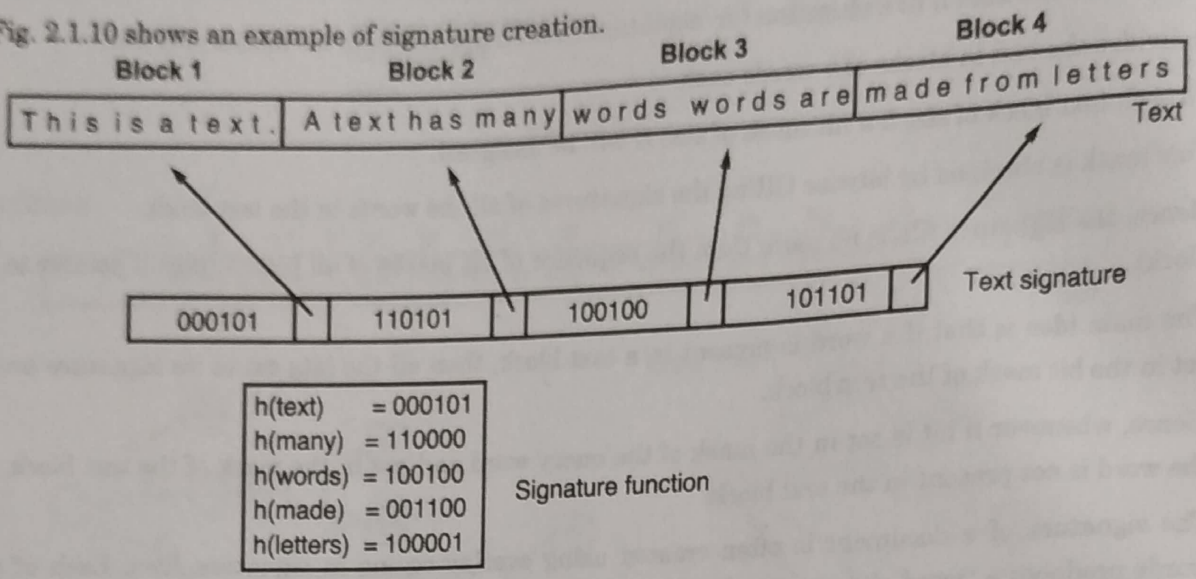


Fig. 2.1.10 : Signature creation

2.1.5 Scatter Storage or Hash Addressing

GQ. Explain Scatter storage or hash addressing concept. **(5 Marks)**

- One file structure which does not relate very well to the ones mentioned before is known as *Scatter Storage*.
- The technique by which the file structure is implemented is often called *Hash Addressing*. Its underlying principle is appealingly simple.
- Given that we may access the data through a number of keys K_i, then the address of the data in store is located through a key transformation function f which when applied to K_i evaluates to give the address of the associated data.
- We are assuming here that with each key is associated only one data item. Also for convenience we will assume that each record (data and key) fits into one location, whose address is in the image space of f.
- The addresses given by the application of f to the keys K_i are called the hash addresses and f is called a hashing function. Ideally f should be such that it spreads the hash addresses uniformly over the available storage. Of course this would be achieved if the function were one-to-one.
- Unfortunately this cannot be so because the range of possible key values is usually considerably larger than the range of the available storage addresses.

- Therefore, given any hashing function we have to contend with the fact that two distinct keys K_i and K_j are likely to map to the same address $f(K_i)$ $(= f(K_j))$. Before I explain some of the ways of dealing with this I shall give a few examples of hashing functions.

Let us assume that the available storage is of size 2[m] then three simple transformations are as follows :

(1) If K_i is the key, then take the square of its binary representation and select m bits from the middle of the result;

(2) Cut the binary representation of K_i into pieces each of m bits and add these together. Now select the m least significant bits of the sum as the hash address;

(3) Divide the integer corresponding to K_i by the length of the available store 2[m] and use the remainder as the hash address.

2.2 SEARCHING TECHNIQUES

GQ. List and explain various searching Techniques. **(6 Marks)**

- The search method is a method for locating pertinent information in information systems.
- Either an internal or online information system may be used.
- An in-house information system is one that stores data for retrieval within the boundaries of an organisation.
- An online information system is one in which communication technology is used to access electronic information sources that are stored remotely.
- The majority of online information systems are World Wide Web (WWW) interoperable and available online.
- The internal information system may have both written and electronic information sources. As a result, the methods for storing information and doing searches are distinct. These two facets of information retrieval and storage will be discussed.
- Scanning the text progressively is recommended while looking for a basic query. Finding instances of a pattern in a text includes sequential or online text searching.
- When the text is short and the text collection is highly volatile or the index space overhead cannot be supported, online searching is the only option.
- The search algorithm on an inverted index has three steps :
  (1) Vocabulary Search (2) Retrieval of occurrence (3) Manipulation of occurrences
- To expedite the search, single-word queries can be searched using any appropriate data structure, such as hashing, attempts, or B-trees.
- The first two give $O(m)$ search cost.
- However, merely storing the words in lexicographic order saves space and performs at a very high level.

- Since the word can be binary searched at $O(\log n)$ cost.
- Binary search, attempts, or B-trees can also be used to solve prefix and range queries; however, hashing cannot. If the question is made up of only one or two words, the process is completed by returning a list of instances.
- With inverted indexes, context queries are more challenging to resolve.
- To create a list for each element, each must be searched independently.
- The next step is to traverse the lists of all the items in sync to locate instances where all the words (for a sentence) appear in order or are sufficiently close together (for proximity). If one list is significantly shorter than the others, binary searching its elements into the longer lists could be preferable to doing a linear merging.
- Since the queries require the position information, block traversal is required if block addressing is used. It is then preferable to intersect the lists to find the blocks containing all the words that were searched, after which you can sequentially search for the context query in those blocks. At block boundaries, caution must be taken because they have the potential to split a match.
- Every search strategy relies on comparing the query to the stored documents. Sometimes, examining the query language will help you understand the distinctions made between various search strategies.
- The type of search technique is frequently determined by the query language. For instance, a Boolean search is typically required by query languages that allow search queries to be stated in terms of logical keyword combinations. This search compares the query and the documents logically to arrive at its results.

2.2.1 Boolean Search

GQ. Explain the Boolean search technique in detail. **(6 Marks)**

- A Boolean search strategy returns documents that match the query as "true." This formula only makes sense if the queries are concatenated using the standard logical connectives AND, OR, and NOT and written as index terms.
- For example, the query $Q = (K_1 \text{ AND } K_2) \text{ OR } (K_3 \text{ AND } (\text{NOT } K_4))$.
- All documents indexed by K_1 and K_2 as well as all documents indexed by K_3 but not indexed by K_4 will be retrieved by the Boolean search.
- Some Boolean search engines allow users to narrow or extend their search by providing them with access to a structured lexicon.
- For example, in the tree structure in Fig. 2.2.1 the keyword K_1^1 is contained in the more general keyword K_1^0, but it can also be split up into the 4 more precise keywords K_1^2, K_2^2, K_3^2, and K_4^2.
- Using the inverted file is a straight forward technique to implement the Boolean search.

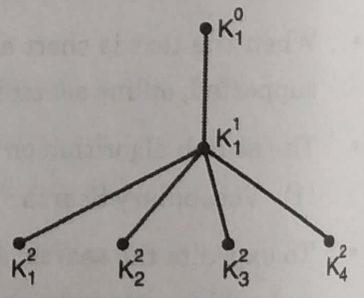


Fig. 2.2.1

- For example

  K_1 -list : D_1, D_2, D_3, D_4

  K_2 -list : D_1, D_2

  K_3 -list : D_1, D_2, D_3

  K_4 -list : D_1

  and $Q = (K_1 \text{ AND } K_2) \text{ OR } (K_3 \text{ AND } (\text{NOT } K_4))$

  Result : {D_1, D_2, D_3}

- A Boolean search that only supports AND logic and takes into account the precise number of phrases the query shares with a document is a modest modification of the complete Boolean search.
- The coordination level is now referred to as this number.
- The search approach is frequently referred to as simple matching.
- Since we can have multiple papers at any level, the documents are considered to be partially prioritised by the coordination levels.

  e.g. query $Q = K_1$ AND K_2 AND K_3
- We obtain the following ranking :
- Co-ordination level

  3 D_1, D_2

  2 D_3

  1 D_4

☞ Matching Functions

GQ. What is matching function? Explain its working. **(6 Marks)**

- A matching function is frequently used to construct search techniques.
- In contrast to association measures, which are applied to items of the same sort, this function matching measure examines the relationship between a query and a document or cluster profile.
- The two functions are equivalent mathematically; the only distinction is in how they are interpreted.
- There are numerous literature examples of matching functions.
- The one linked to the straightforward matching search approach is conceivably the easiest.
- If M is the matching function, D the set of keywords representing the document, and Q the set representing the query, then

$$M = \frac{2|D \cap Q|}{|D| + |Q|}$$

- The document and query are represented as numerical vectors in t-space, that is $Q = (q_1, q_2, \ldots, q_t)$ and $D = (d_1, d_2, \ldots, d_t)$ where q_i and d_i are numerical weights associated with the keyword i.

- The cosine correlation is

$$t = \frac{\sum_{i=1}^{t} q_i d_i}{\left(\sum_{i=1}^{t} (q_i)^2 \sum_{i=1}^{t} (d_i)^2 \right)^{\frac{1}{2}}}$$

- or, in the notation for a vector space with a Euclidean norm,

$$r = \frac{(Q, D)}{||Q|| \; ||D||} = \text{cosine } \theta$$

2.2.2 Sequential Search, Serial Search

GQ. Explain Sequential search or serial search in detail. (6 Marks)

- Suppose there are N documents Di in the system, then the serial search proceeds by calculating values $M(Q, D_i)$ the set of documents to be retrieved is determined.
- There are two ways of doing this :
  (1) The matching function is given a suitable threshold, retrieving the documents above the threshold and discarding the ones below.
  If T is the threshold, then the retrieved set B is the set
  $\{D_i \mid M(Q, D_i) > T\}$.
  (2) The documents are ranked in increasing order of matching function value.
- A rank position R is chosen as cut-off and all documents below the rank are retrieved so that $B = \{D_i \mid r(i) < R\}$ where r(i) is the rank position assigned to D_i.

The hope in each case is that the relevant documents are contained in the retrieved set.

- **Disadvantage :** Arbitrary Threshold Value

2.2.3 Cluster-based Retrieval

GQ. Explain the cluster –based retrieval in detail. (6 Marks)

Cluster representatives

- In contrast, a serial search requires that we match queries with every document in the file. We need to be able to match queries with clusters while searching a file that has been clustered.
- A type of profile hereafter referred to as a cluster representative represents clusters for this purpose.
- A cluster representative should be set up such that an incoming query is directed to the cluster that has the query-relevant documents. There are numerous "reasonable" methods for describing clusters.
- An illustration of a very basic cluster representation. We can represent each cluster by a graph if we assume that the clusters were created using a cluster technique based on a dissimilarity measure.

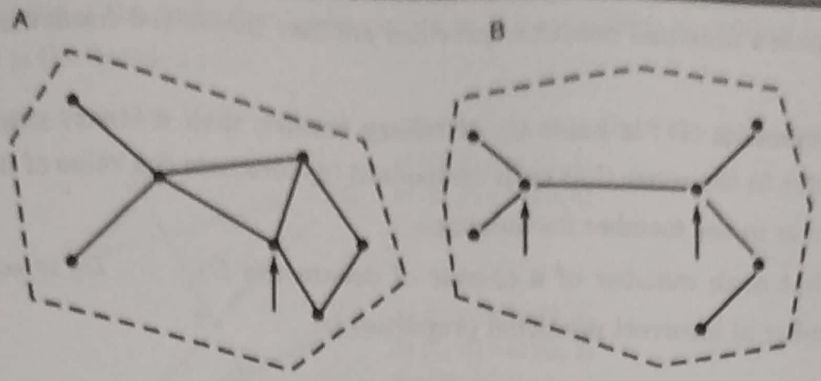


Fig. 2.2.2

- The term "maximally linked document" is used to identify the document in a cluster that is related to the greatest number of other documents. a representation strategy that 'averages' the cluster members' descriptions in some way.
- If $\{D_1, D_2, \ldots, D_n\}$ are the documents in the cluster and each Di is represented by a numerical vector $(d_1, d_2, \ldots, d_t)$ then the centroid C of the cluster is given by

$$C = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i}{|| D_i ||}$$

where $|| D_i ||$ is usually the Euclidean norm, i.e.

$$|| D_i || = \sqrt{d_1^2 + d_2^2 + \ldots + d_t^2}$$

- Typically, binary vectors rather than numerical ones are used to represent the documents.
- Let D_i now be a binary vector, such that a 1 in the j^{th} position indicates the presence of the j^{th} keyword in the document and a 0 indicates the contrary.
- The cluster representative is now derived from the sum vector

$$S = \sum_{i=1}^{n} D_i$$

- Let $C = (c_1, c_2, \ldots c_t)$ be the cluster representative and $[D_i]_j$ the j^{th} component of the binary vector D_i, then two methods are:

1. $$c_j = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} [D_i]_j > 1 \\ 0 \text{ otherwise} \end{cases}$$

or

2. $$c_j = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} [D_i]_j > \log_2 n \\ 0 \text{ otherwise} \end{cases}$$

- The idea of a cluster's maximal predictor provides another theoretical framework for creating cluster representatives.
- If a cluster of documents (D_i) is made up of binary vectors, then a binary cluster representing that cluster is a predictor in the sense that each component (c_i) forecasts the value of that characteristic that will most likely occur in the member documents.
- If one assumes that each member of a cluster of documents $D_1, \ldots, D_n$ is equally likely then the expected total number of incorrect predicted properties is,

$$\sum_{i=1}^{n} \sum_{j=1}^{t} ([D_i]_j - c_j)^2$$

This can be rewritten as

$$\sum_{i=1}^{n} \sum_{j=1}^{t} ([D_i]_j - D_{\cdot j})^2 + n \sum_{j=1}^{t} ([D_i]_j - c_j)$$

Where

$$D_{\cdot j} = \frac{1}{n} \sum_{i=1}^{n} [D_i]_j$$

- The expression (*) will be minimized, thus maximizing the number of correct predictions, when $C = (c_1, \ldots, c_t)$ is chosen in such a way that

$$\sum_{j=1}^{t} ([D_i]_j - c_j)^2$$

- is a minimum. This is achieved by

3. $$c_j = \begin{cases} 1 & \text{if } D_{\cdot j} > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

☞ **Cluster-based retrieval**

- The cluster hypothesis, which asserts that closely related documents are frequently relevant to the same queries, serves as the basis for cluster-based retrieval.
- Clustering selects documents that are closely related and gathers them into a cluster.
- Assuming our classification of documents is hierarchical, the following is a straightforward search approach. In the example, node 0 is where the search begins.
- The next step is to evaluate a matching function at the nodes directly below node 0, in this case, nodes 1 and 2 in the example. Along the tree, this pattern is repeated.
- A decision rule controls the search and determines which node to extend further at each stage based on a comparison of the values of a matching function.
- The need for a stopping rule that stops the search and requires a retrieval is also essential.

- In Fig. 2.2.3, expanding the node that corresponds to the maximum matching function value attained within a final set is the decision rule.
- The rule for stopping is to stop if the current maximum is lower than the prior maximum.

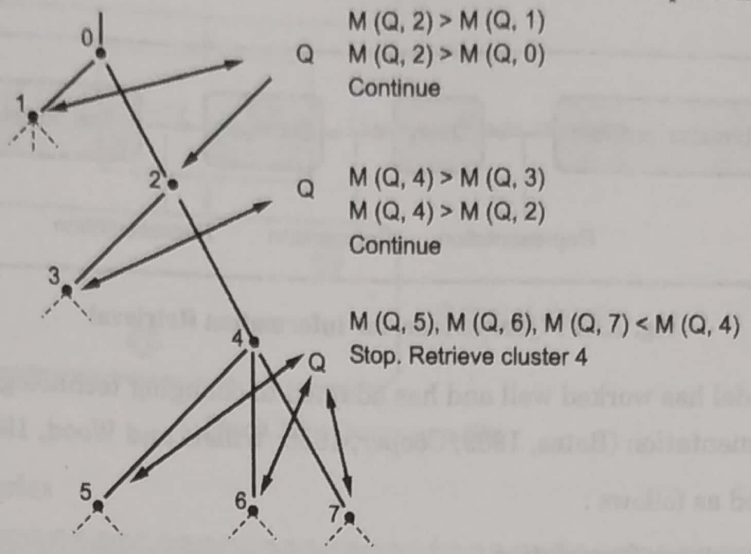


Fig. 2.2.3

- A generalisation of this search would allow retrieval of several clusters by allowing the search to go down multiple tree branches. The decision rule and stopping rule will inevitably be a little more difficult.
- The primary distinction is the requirement for backtracking. Top-down searches could be used to describe the aforementioned tactics.
- A bottom-up search is one that begins at one of the tree's terminal nodes and works its way upward toward the tree's root.
- It will proceed in this manner through a series of nested clusters of escalating size. We only need a halting rule, which might be as simple as a cut-off; a decision rule is not necessary.
- The starting node's document is represented by this cluster, and a typical search would look for the largest cluster that doesn't go over the cut-off size.
- The group of documents it contains are retrieved after this cluster has been located. It is vital to have an idea of which terminal node would be best for a given request before starting the search in response to one. It is common to discover that a user is looking for papers that are comparable to one they already know about and that relate to their request. So, one can start a bottom-up search using this "source" document. In order to assess the effectiveness and efficiency of bottom-up searches.

2.3 QUERY LANGUAGES

GQ. Explain the query processing in IR. **(4 Marks)**

- An information retrieval (IR) query language is a query language used to create search index queries.
- A query language is defined formally in a context-free grammar (CFG) and can be used by users in textual, visual/UI, or speech form.

- In vertical search engines, advanced query languages are frequently defined for professional users to give them more control over query formulation.
- The classic model depicted in Fig. 2.3.1 dominates information retrieval systems and research (Belkin and Croft, 1987).

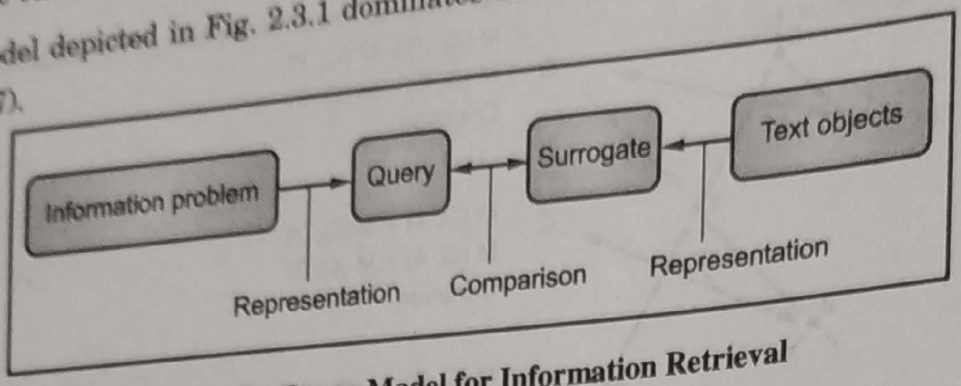


Fig. 2.3.1 : Query Model for Information Retrieval

- Even though this model has worked well and has adapted to changing technologies, there are still many issues with its implementation (Bates, 1989; Cooper, 1988; Willett and Wood, 1989).
- These are summarised as follows :
  (1) User challenges in query formulation
  (2) Too many matches and no returns
  (3) The importance of query components and term dependencies varies.
  (4) Adaptation to various types of objects, such as images and sounds
  (5) Finding appropriate surrogate representations for objects.
- There is a universe of objects to be searched in this model, with surrogates representing the objects (such as indexing terms or keywords).
- When a searcher enters a query, the information retrieval system matches the query against the surrogate representations and presents the matching objects to the user for consideration.
- Users then examine the retrieved objects and choose those that are of interest.
- Browsing can be defined as an interactive search activity in which the user determines the direction of the search based on immediate feedback from the system being browsed. Regardless of the underlying system structure, most users of most information retrieval systems exhibit browsing behaviour.
- Many readers, for example, are familiar with their own behaviour when searching with query-based systems, in which the results of their queries are used to create their next query.
- A query-based system, as shown in Fig. 2.3.1, can also be viewed as defining a subset of objects that the user examines.
- The values of object attributes are used to select objects from a loosely structured database (Fig. 2.3.2).

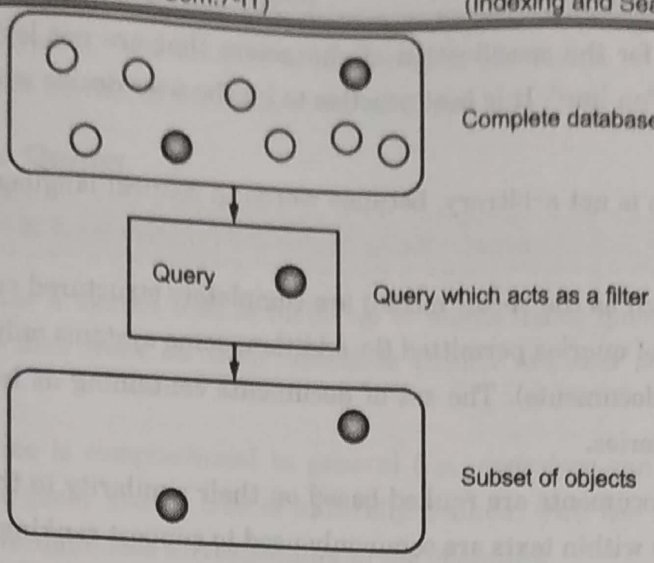


Fig. 2.3.2 : Query as a filter

2.3.1 Types of Queries

GQ. Enlist and explain the various types of queries in IR. **(6 Marks)**

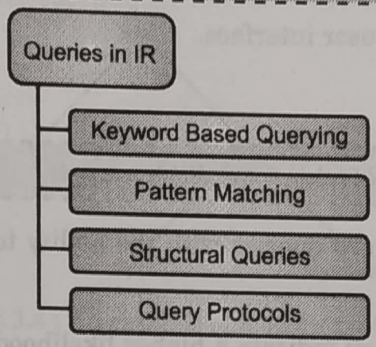


Fig. 2.3.3 : Types of queries in IR

2.3.2 Keyword Based Querying

GQ. What is Keyword based Querying. Explain with example. **(5 Marks)**

- A query is the expression of a user's information requirement.
- In its most basic form, a query is made up of keywords, and the documents that contain those keywords are searched for.
- Keyword queries are popular because they are simple to express and allow for quick ranking. Thus, a query can (and often is) just a word, though it can also be a more complex combination of operations involving multiple words.

2.3.2(A) Single-Word Queries

GQ. What do you mean by Single-word query? Explain with suitable example. **(4 Marks)**

- A word is normally defined in a straightforward manner. A word is a sequence of letters surrounded by separators, and the alphabet is divided into letters and separators.

- More complex models allow for the specification of characters that are not letters but do not split a word, such as the hyphen in "on-line". It is best practise to let the user decide what a letter is and what a separator is.
- The text's division into words is not arbitrary, because words in natural language carry a great deal of meaning.
- As a result, many models (such as the vector model) are completely structured on the concept of words and words are the only type of queries permitted (in addition, some systems only allow a limited set of words to be extracted from documents). The set of documents containing at least one of the query's words is the result of word queries.
- Furthermore, the resulting documents are ranked based on their similarity to the query. Two common statistics on word occurrences within texts are commonly used to support ranking.
- The first is called "term frequency," and it counts how many times a word appears within a document. The second method, known as "document frequency," is based on counting the number of documents in which a word appears.
- Furthermore, the exact position of the word in the text may be required. This could be useful for highlighting each occurrence in the user interface.

2.3.2(B) Context Queries

GQ. What do you mean by context queries? Explain with suitable example. **(4 Marks)**

- Many systems supplement single-word queries with the ability to search for words in a given context, that is, near other words.
- Words that appear close together may indicate a higher likelihood of relevance than words that appear separately. For example, we may want to form phrases of words or and words that are close together in the text.

Phase

- A phrase is a collection of single-word queries. A sequence of words is an occurrence of the phrase.
- **For example**, you could look up the words "enhance" and "retrieval" in the dictionary. In phrase queries, it is generally assumed that the separators in the text do not have to be the same as those in the query (e.g., two spaces versus one space), and that uninteresting words are ignored entirely.
- For instance, the previous example could match in a text such as "...enhance the retrieval...". Although this feature is very useful in most cases, not all systems implement it.

Proximity

- The proximity query is a more relaxed variant of the phrase query. In this case, a sequence of single words or phrases is provided, as well as the maximum allowable distance between them.
- **For example**, in the preceding example, the two words should appear within four words, and thus a match could be "...enhance the power of retrieval..."

- Depending on the system, this distance is measured in characters or words. It is not necessary for the words and phrases to appear in the same order as in the query.

2.3.2(C) Boolean Queries

GQ. What do you mean by boolean query? Explain with suitable example. **(4 Marks)**

- A boolean query has a syntax that is made up of atoms (basic queries) that retrieve documents and boolean operators that work on their operands (which are sets of documents) and return sets of documents.
- Because this scheme is compositional in general (i.e., operators can be composed over the results of other operators), a query syntax tree is naturally defined, with the leaves corresponding to the basic queries and the internal nodes corresponding to the operators.
- The query syntax tree is based on an algebra over sets of documents (the query's final answer is also a set of documents).
- This is similar to the syntax trees of arithmetic expressions, where the leaves are numbers and variables and the internal nodes are operations.

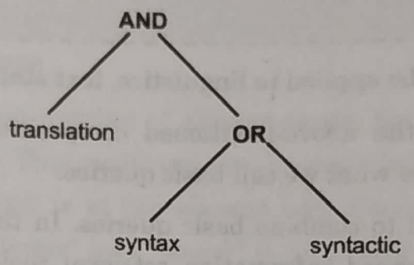


Fig. 2.3.4 : An example of a query syntax tree.

- It will retrieve all the documents which contain the word "translation" as well as either the word "syntax" or the word "syntactic".
- The operators most commonly used, given two basic queries or boolean sub-expressions e_1 and e_2, are:

OR

The query (e_1 OR e_2) selects all documents which satisfy e_1 or e_2. Duplicates are eliminated.

AND

The query (e_1 AND e_2) selects all documents which satisfy both e_1 and e_2.

BUT

- The query (e_1 BU T e_2) selects all documents which satisfy e_1 but not e_2. Notice that classical boolean logic uses a \NOT " operation, where (NOT e_2) is valid whenever e_2 is not.
- In this case all documents not satisfying e_2 should be delivered, which may retrieve a huge amount of text and it is probably not what the user wants. The BU T operator, instead, sets the universe of retrievable elements to the result of e_1.

2.3.2(D) Natural Language

GQ. What do you mean by natural language query? Explain with suitable example. (4 Marks)

- By pushing the fuzzy boolean model even further, the distinction between AND and OR could be completely blurred, so that a query becomes simply an enumeration of words and context queries of interest to the user, and all the documents matching some query are retrieved, with more weight given to those matching more parts of the query.
- The negation can be handled by allowing the user to express that certain words are undesirable, in which case the documents that contain them are penalised in the ranking computation.
- A threshold can be set to prevent documents with extremely low weights from being retrieved.
- We have completely eliminated any reference to Boolean operations under this scheme and entered the field of natural language queries.
- In fact, Boolean queries can be thought of as a simplified abstraction of natural language queries.

2.3.3 Pattern Matching

GQ. Explain the term: Pattern Matching. (5 Marks)

- These data retrieval queries can be applied to linguistics, text statistics, and data extraction.
- Their output can be fed into the above-mentioned composition mechanism to form phrases and proximity queries, which comprise what we call basic queries.
- Boolean expressions can be used to combine basic queries. In this sense, we can consider these data retrieval capabilities to be enhanced information retrieval tools. However, ranking the result of a pattern matching expression is more difficult.
- A pattern is a collection of syntactic elements that must be present in a text segment. The pattern is said to "match" those segments that satisfy the pattern specifications.
- We're looking for documents that contain segments that match a specific search pattern. Each system allows you to specify different types of patterns, ranging from very simple (for example, words) to quite complex (such as regular expressions).
- The more powerful the set of patterns allowed, the more involved queries the user can formulate and, in general, the more complex the search implementation.

The most common patterns are :

(1) **Words :** a string (sequence of characters) which must be a word in the text. This is the most basic pattern.

(2) **Prefixes :** a string which must form the beginning of a text word. For instance, given the prefix "comput" all the documents containing words as "computer", computation", "computing", etc. are retrieved.

(3) **Suffixes :** a string which must form the termination of a text word. For instance, given the suffix "ters" all the documents containing words as "computers", "testers", "painters", etc. are retrieved.

(4) Substrings : a string which can appear within a text word. For instance, given the substring "tal" all the documents containing words such as "coastal", "talk", "metallic", etc. are retrieved. This query can be restricted to and the substrings inside words, or it can go further and search the substring anywhere in the text (in this case the query is not restricted to be a sequence of letters but can contain word separators). For instance, a search for "any flow" will match in the phrase "...many flowers...".

(i) Ranges : are a pair of strings that match any word that is lexicographically between them. Normally, alphabets are sorted, which results in an order in the strings known as lexicographical order (this is indeed the order in which words in a dictionary are listed). For example, the range "held" and "held" will return strings like "hoax" and "hissing."

(ii) Allowing for errors : a word plus an error threshold. This search pattern returns all text words that are "similar" to the specified word. The concept of similarity can be defined in a variety of ways.

(1) Edit Distance

The minimum number of character insertions, deletions and replacements needed to make them equal. For instance, the edit distance between "color" and "colour" is one, while the edit distance between "survey" and "surgery" is two.

(2) Maximum allowed edit distance

The query specifies the maximum number of allowed errors for a word to match the pattern (i.e. the maximum allowed edit distance). This model can also be extended to search substrings (not only words), retrieving any text segment which is at the allowed edit distance to the search pattern. Under this model, if a typing error splits "flower" into "flower" it could still be found with one error, while in the restricted case of words it could not(since neither "flo" or "wer" are at edit distance 1 to "flower").

(3) Regular expressions : some text retrieval systems allow searching for regular expressions. A regular expression is a rather general pattern built up by simple strings (which are meant to be matched as substrings) and the following operators.

(i) Union : if e1 and e_2 are regular expressions, then (e_1|e_2) matches what e1 or e_2 matches.

(ii) Concatenation : if e1 and e_2 are regular expressions, the occurrences of (e_1 e_2) are formed by the occurrences of e1 immediately followed by those of e_2 (therefore simple strings can be thought of as a concatenation of their individual letters).

(iii) Repetition : if e is a regular expression, then (e) matches a sequence of zero or more contiguous occurrences of e.

For instance, consider a query like "pro (blem | tein) (s |∈) (0 | 1 | 2)" (where ∈ denotes the empty string). It will match words such as "problem02" and "proteins".

(iv) Extended patterns : It is common to represent some common cases of regular expressions with a more user-friendly query language. Extended patterns are subsets of regular expressions that have a simpler syntax. Internally, the retrieval system can convert extended patterns into regular expressions or search them using specific algorithms. Because each system supports its own set of extended patterns, no formal definition exists.

Some examples found in many new systems are as follows :

- Character classes, i.e. some pattern positions that match with a set of characters This includes features like case-insensitive matching, the use of character ranges (e.g. specifying that some characters must be digits), complements (e.g. some characters must not be letters), enumeration (e.g. a character must be a vowel), and wild cards (i.e. some pattern position matches with anything).
- Conditional expressions, in which a portion of the pattern appears or does not appear.
- Wild characters that match any sequence in the text, such as any word that begins with "flo" and ends with "ers," which matches both "flowers" and "flounders."
- Combinations that allow some parts of the pattern to match exactly while others have errors

2.3.4 Structural Queries

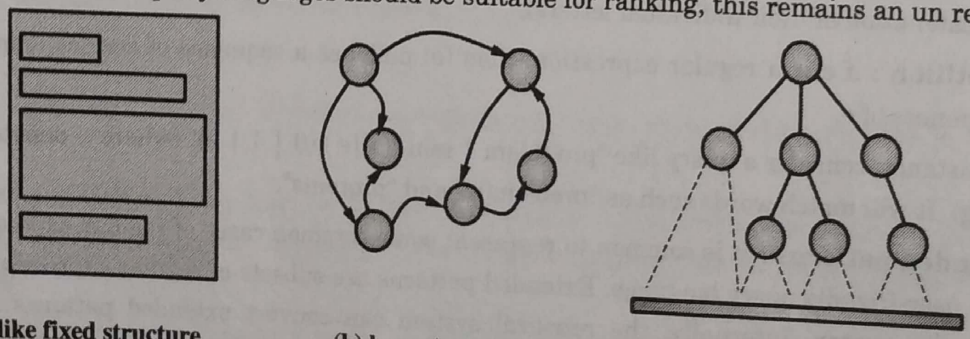
GQ. Explain the structural queries with suitable example. **(4 Marks)**

GQ. Explain the types of structural queries. **(6 Marks)**

GQ. Write a short note on : **(6 Marks)**

(1) Fixed Structured queries (2) Hypertext queries (3) Hierarchical queries

- Combining contents and structure in queries enables the creation of extremely powerful queries that are far more expressive than each query mechanism alone. The retrieval quality of textual databases can be improved by using a query language that integrates both types of queries.
- This mechanism is built on top of the basic queries to select a set of documents that meet certain content constraints (expressed using words, phrases or patterns that the documents must contain).
- In addition, some structural constraints can be expressed in the documents through containment, proximity, or other restrictions on structural elements (e.g., chapters, sections, etc.). Boolean queries can be constructed on top of structural queries to combine the sets of documents returned by those structural queries.
- The structural queries are the tree's leaves in the boolean syntax tree (recall the example in Fig. 2.3.4). Structured queries, on the other hand, can have a complex syntax. This section is divided into sections based on the types of structures found in text databases. Fig. 2.3.5 depicts them.
- Although structured query languages should be suitable for ranking, this remains an un resolved issue.



(a) form-like fixed structure **(b) hypertext structure and** **(c) hierarchical structure**

Fig. 2.3.5 : The three main structures

- In what follows, it is critical to distinguish between the structure of a text and what can be questioned about that structure. Natural language texts can have any structure they want.
- However, different models allow you to query only a subset of the true structure. When we say that the allowed structure is restricted in some way, we mean that only the aspects that follow this restriction can be queried, even though the text may contain more structural information.

2.3.4(A) Fixed Structure

- Text structure has traditionally been quite restricted. The documents, like a filled form, had a fixed set of fields.
- Each field contained some text. Some fields were not present in all documents, but they could rarely appear in any order or repeatedly across the document, or the document could contain text that was not classified under any field. It was forbidden for them to nest or overlap.
- The retrieval activity permitted on them was limited to specifying that a given basic pattern be found only in a specific field. This model is used by the majority of current commercial systems. When the text collection has a fixed structure, this model makes sense.
- A mail archive, for example, could be thought of as a collection of emails, each with a sender, a receiver, a date, a subject, and a body field. Thus, the user can search for emails with the subject "football" that he sent to a specific person. However, the model is insufficient to represent the hierarchical structure found in an HTML document, for example.

2.3.4(B) Hypertext

- In terms of structuring power, hypertexts most likely represent the inverse trend. A hypertext is a directed graph in which the nodes contain text and the links represent connections between nodes or positions within nodes.
- Since the explosion of the Web, which is a massive hypertext-like database spread across the globe, hypertexts have received a lot of attention.
- However, hypertext retrieval began as a purely navigational activity. That is, the user had to manually navigate the hypertext nodes by following links to find what he or she was looking for.
- The hypertext could not be queried based on its structure. Even on the Web, one can search for nodes based on their text contents, but not on their connection structure.

2.3.4(C) Hierarchical Structures

- An intermediate structuring model which lies between fixed structure and hypertext is the hierarchical structure. This represents a recursive decomposition of the text and it is a natural model for many text collections (e.g. books, articles, legal documents, structured programs, etc.). Fig. 2.3.6 shows an example of such structure.
- On the other hand, the simplification from hypertext to a hierarchy allows the use of faster algorithms to solve queries. As a general rule, the more powerful the model, the less efficiently it can be implemented.

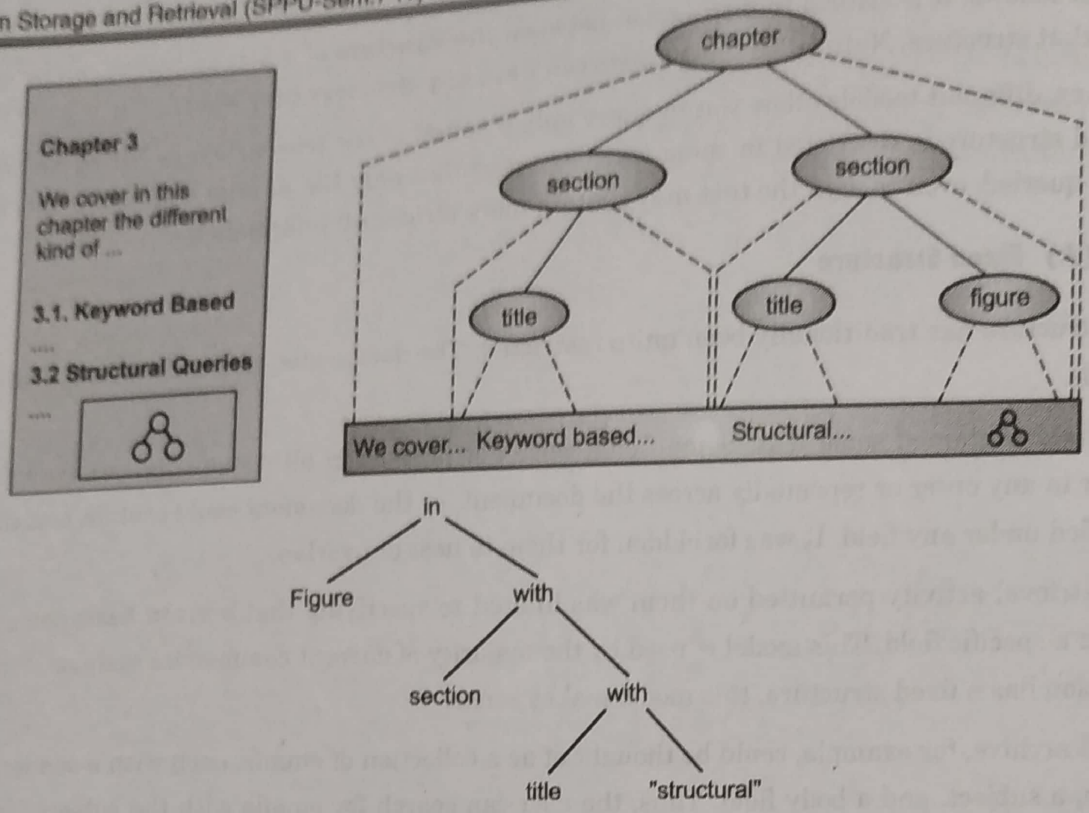


Fig. 2.3.6 : An example of a hierarchical structure : the page of a book, it schematic view, and a parsed query to retrieve the figure

2.3.4(C.1) Hierarchical Structures - A Sample of Hierarchical Models

- **PAT Expressions -** are based on the same index of the text, i.e. there is no structure index. Because the structure is assumed to be marked in the text by tags (as in HTML), it is defined in terms of initial and final tags.
- This enables a dynamic scheme in which the structure of interest is not fixed but can be determined at query time (because the tags do not need to be specially designed to be tags, for example, one can de ne that the end-of-lines are the marks in order to de ne a structure of lines). This also allows for a very efficient implementation with no additional structure space overhead.
- Each expression of the initial and final tags denotes a region, which is a collection of contiguous text areas. Externally computed regions can also be used. The areas of a region, however, cannot nest or overlap, which is quite restrictive. There are no restrictions on different regions' areas.
- The algebra has the disadvantage of combining regions and sets of text positions, which are incompatible and necessitate complex conversion semantics. For example, if the result of a query will produce overlapping areas (which cannot be predicted in advance), the result is converted to positions.
- Furthermore, dynamic definition of regions is flexible, but it requires that the structure be expressed using tags (also known as "markup"), which does not occur in some structured programming languages.

Overlapped Lists

- can be viewed as a progression from PAT Expressions The model allows regions' areas to overlap but not nest. This elegantly solves the problems associated with combining regions and sets of positions.

- This model's implementation can also be very efficient. However, it is unclear whether overlapping is a good way to capture the structural properties of information in practise.
- A new proposal allows the structure to nest and overlap, demonstrating that the majority of interesting operators can still be implemented.

☞ Lists of References

- Is an attempt to standardise the definition and querying of structured text through the use of a common language. We limit our attention to the subset of our interest because the language goes beyond querying structured text.
- Answers to queries are viewed as collections of references." A reference is a pointer to a database region. Because all of these are lists of references, this elegantly integrates answers to queries and hypertext links.

☞ Proximal Nodes

- Tries to strike a balance between expressiveness and efficiency It does not define a specific language, but rather a model in which it is demonstrated that a number of useful operators can be included while still achieving high efficiency.
- The structure is predetermined and hierarchical. However, many independent structures can be defined on the same text, each of which is a strict hierarchy with overlaps between areas of different hierarchies.
- A query can connect different hierarchies, but it only returns a subset of the nodes in one hierarchy (i.e. nested elements are allowed in the answers, but no overlaps). Text matching queries are represented as nodes in a special "text hierarchy."

☞ Tree Matching

- Tree matching is founded on a single primitive: tree inclusion The concept of tree inclusion is to consider both the database structure and the query (a pattern on structure) as trees, and to embed the query into the database while respecting the hierarchical relationships between query nodes.
- Ordered inclusion requires the embedding to respect the query's siblings' left-to-right relationships, whereas unordered inclusion does not. The query's leaves can be both structural elements and text patterns, which means that the pattern must be present in the leaf's ancestor.
- Simple queries return the roots of the matches, and the language is enhanced with Prolog-like variables that can be used to express equality requirements between parts of the matched substructure and retrieve another part of the match, not just the root.
- Logical variables are also used to simulate tuples and join capabilities, as well as for query union and intersection.

2.4 IR MODELS : BASIC CONCEPTS

Taxonomy of Information Retrieval Models (4 Marks)

GQ. Explain the taxonomy of information retrieval models in details.

- 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, d_j) 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.

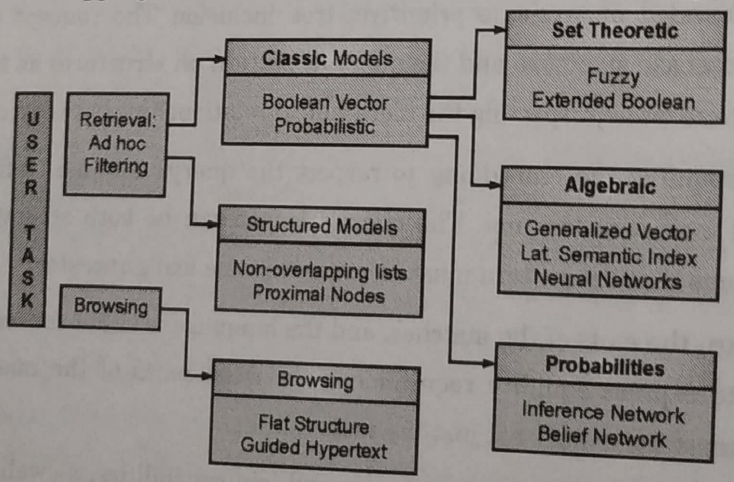


Fig. 2.4.1 : Information Retrieval Models

- The classical information retrieval models are depicted in Fig. 2.4.1. The following sections elaborate on the classic models.

Basic models of information retrieval a brief overview

- A mathematical model of information retrieval guides the implementation of information retrieval systems.
- Only the matching process is automated in traditional information retrieval systems, which are 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 retrieval systems.

2.4.1 Boolean Model

GQ. Explain the Boolean model in detail. (6 Marks)

UQ. Explain Boolean model in detail. (SPPU - Q. 3(a), May 16, 5 Marks)

UQ. Compare Boolean model and vector model . Explain how vector model can be used to retrieve partial matching document. (SPPU - Q. 3(a), April 17, 6 Marks)

UQ. Compare Boolean and vector model. (SPPU - Q. 2(a), May 19, 6 Marks)

- 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. 2.4.2, Fig. 2.4.3 and Fig. 2.4.4).
- Grey areas represent the set of documents that can be retrieved.

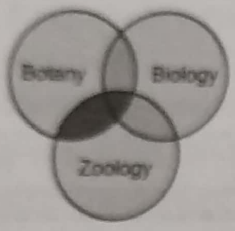


Fig. 2.4.2 : Botany AND Zoology

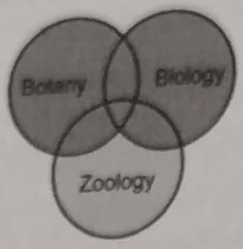


Fig. 2.4.3 : Botany OR Zoology

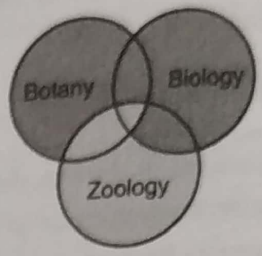


Fig. 2.4.4 : (Botany OR Biology) AND NOT (Botany AND Zoology)

- 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

(1) **D** – A set of words, i.e., the indexing terms present in a document. Here, each term is either present (1) or absent (0).

(2) **Q** – A Boolean expression, where terms are the index terms and operators are logical products – AND, logical sum – OR and logical difference – NOT.

(3) **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 :

(4) **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 term **"economic"**.
- Now, what would be the result after combining terms with Boolean AND Operator ? It will define a document set that is smaller than or equal to the document sets of any of the single terms. For example, the query with terms **"social"** and **"economic"** will produce the documents set of documents that are 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 define a document set that is bigger than or equal to the document sets of any of the single terms. For example, the query with terms **"social"** or **"economic"** will produce the documents set of documents that are indexed with either the term **"social"** or **"economic"**. In other words, document set with the union of 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 be 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.

2.4.2 Vector Model

GQ. Explain Vector Model in details. **(6 Marks)**

UQ. Explain Vector model in detail. **(SPPU - Q. 3(a), Dec. 18, 5 Marks)**

GQ. Find the similarity of the following query with documents - D1, D2, D3 using vector model.

Query	Keywords	
Q	Mouse, dog	
Document	Text	Terms
D1	Mouse mouse bee	mouse bee
D2	Dog bee dog hog Hog dog mouse dog	mouse bee dog hog

UQ. Find the similarity of following query with D1, D2, D3 using vector model. **(SPPU - 9, 2(a), May 17, 5 Marks)**

Query	Keywords
Q	ant, dog
document	Text
D1	ant ant bee
D2	dog bee dog hog dog ant dog
D3	cat gnu dog eel fox

Terms	
ant bee	
ant bee dog hog	
cat dog eel fox gnu	

- 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. 2.4.5 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. 2.4.5.

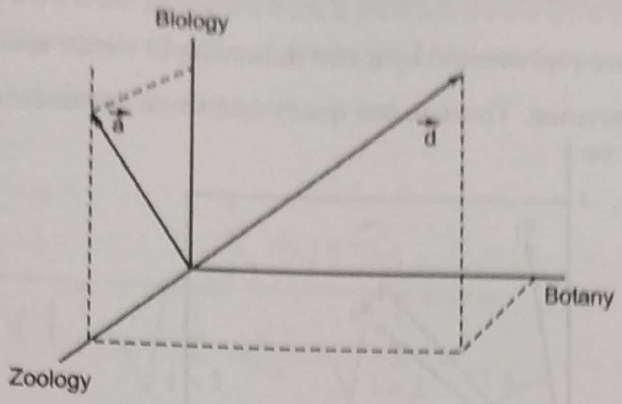


Fig. 2.4.5 : A query and documents representation in the vector space model

- 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) =

$$\sum k = 1mn(dk) \cdot n(qk) \quad \ldots(1)$$

Where $n(vk) = vk \sum k = 1mvK^2$ ...(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 in 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 a model, which is based on Luhn's similarity criterion.
- The similarity criterion formulated by Luhn states, "the more two representations agreed in given elements and their distribution, the higher would be the 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 high dimensional Euclidean space.
- The similarity measure of a document vector to a query vector is usually the cosine of the angle between them.

☞ Vector Space Representation with Query and Document

(6 Marks)

GQ. How to represent the query and document in Vector space. (SPPU - Q. 3(b), May 2018, 5 Marks)

UQ. Justify how vector model is used to retrieve partial matching documents.

- 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.

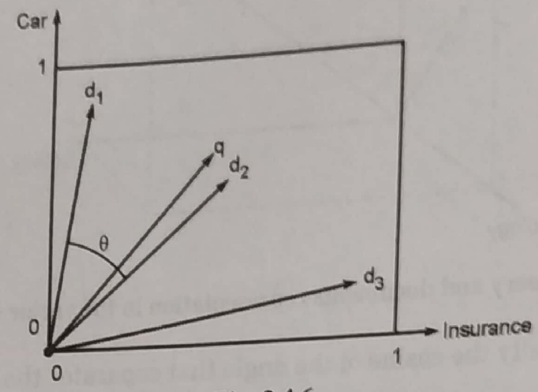


Fig. 2.4.6

- The top ranked document in response to the terms car and insurance will be the document $\mathbf{d_2}$ because the angle between **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 → TF x IDF.
- Compute the similarity between documents and query → Sim (Dj, Q)
- More precise than Boolean model

Idea for TF x IDF→

- **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

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

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

- 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**

$$\text{sim}(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{\left|\vec{d_j}\right| \times \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

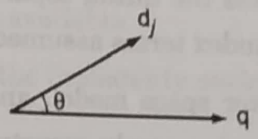


Fig. 2.4.7 : The cosine of θ is adopted as sim (d_j, q)

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.

2.4.3 Probabilistic Model

GQ. What is probability theory ? (2 Marks)
GQ. Explain Probabilistic model. (6 Marks)
GQ. Explain binary independence model in detail. (6 Marks)
UQ. Write a short note on probabilistic model vector model. (SPPU - Q. 3(b), Dec. 18, 4 Marks)
UQ. Write short note on probabilistic model. (SPPU - Q. 2(b), May 2018, 4 Marks)

- 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. 2.4.8 shows an example of the Probabilistic Retrieval model in action.
- 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.

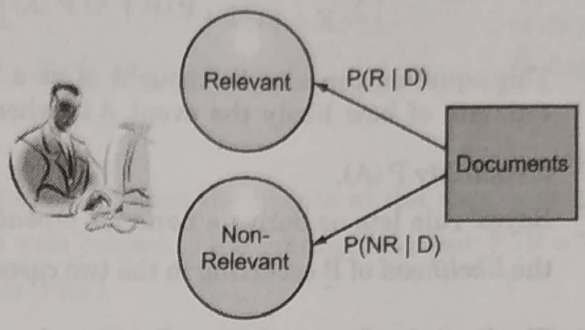


Fig. 2.4.8 : The probabilistic retrieval

- The Bayes Rule, which is also used in the Probability model, is defined as follows.

$$P(R|D) = PDRP(R)P(D) \quad \ldots(3)$$

- 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.

☞ 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* :

$$P(A, B) = P(A \cap B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A) \quad \ldots(4)$$

- 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 :

$$P(\bar{A}, B) = P(B \mid \bar{A})\, P(\bar{A}) \quad \ldots(5)$$

- Probability theory also has a *partition rule*, which says that if an event B can be divided into an exhaustive set of disjoint sub cases, then the probability of B is the sum of the probabilities of the sub cases.

- A special case of this rule gives that : ...(6)

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

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

$$P(A \mid B) = P(B \mid A)\, P(A)$$

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

- 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 how probabilities change :

$$\text{Odds : } O(A) = \frac{P(A)}{P(\bar{A})} = \frac{P(A)}{1 - P(A)} \quad ...(7)$$

☞ Probabilistic Information Retrieval

GQ. Explain the working of probabilistic information retrieval model in brief. **(6 Marks)**

- As we observed that if we have some known relevant and non-relevant documents, then we can straight forwardly start to estimate the probability of a term t appearing in a relevant document P (t | 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 differing at least by how text is tokenized, but perhaps containing fundamentally less information, as when a non-positional index is used).
- Based on these two representations, a system tries to determine how well documents satisfy information needs. In the Boolean or vector space models of IR, matching is done in a formally defined 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 a document has content relevant to the information need.
- Probability theory provides a principled foundation for such reasoning under uncertainty. It is used to 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.

The Probability Ranking Principle

GQ. Explain the Probability Ranking Principle. (4 Marks)

GQ. Explain the 1/0 Loss case in details. (4 Marks)

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|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 called1/0 loss).
- The goal is to return the best possible results as the top k documents, for any value ofk 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: ...(8)

d is relevant iff P (R = 1 | d, q) > P (R = 0 | d, q)

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.

$$C_0 \cdot P(R = 0|d) - C_1 \cdot P(R = 1|d) \leq C_0 \cdot P(R = 0|\underline{d}') - C_1 \cdot P(R = 1|\underline{d}') \quad \ldots(3)$$

- 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.

The Binary Independence Model

(5 Marks)

GQ. What is Binary Independence Model (BIM). Explain its working.

- 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 $\underline{t}$ is present in document d and $x_t = 0$ if $\underline{t}$ is not present in d.
- With this representation, many possible documents have the same vector representation.
- Similarly, we represent $\underline{q}$ by the incidence vector $\vec{q}$ (the distinction between $\underline{q}$ and $\vec{q}$ is less central since commonly $\underline{q}$ is in the form of a set of words).
- "Independence" means that terms are modeled as occurring in documents independently. The model recognizes no association between terms.
- This assumption is far from correct, but it nevertheless often gives satisfactory results in practice; it is the "naive"' assumption of Naive Bayes models.
- Indeed, the Binary Independence Model is exactly the same as the multivariate Bernoulli Naive Bayes model presented in. In a sense this assumption is equivalent to an assumption of the vector space 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 documents contribute to relevance, specifically, we wish to know how term frequency, document frequency, document length, and other statistics that we can compute influence judgments about document relevance, and how they can be reasonably combined to estimate the probability of document relevance.
- 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 other documents.
- This is incorrect: the assumption is especially harmful in practice if it allows a system to return duplicate or near duplicate documents.

- Under the BIM, we model the probability $P(R|\vec{d}, q)$ that a document is relevant via the probability in terms of term incidence vectors $P(R \mid \vec{x}, \vec{q})$. Then, using Bayes rule, we have:

$$P(R=1 \mid \vec{x}, \vec{q}) = \frac{P\left(\vec{x} \mid R=1, \vec{q}\right) P\left(R=1 \mid \vec{q}\right)}{P\left(\vec{x} \mid \vec{q}\right)} \quad \ldots(10)$$

$$P(R=0 \mid \vec{x}, \vec{q}) = \frac{P\left(\vec{x} \mid R=0, \vec{q}\right) P\left(R=0 \mid \vec{q}\right)}{P\left(\vec{x} \mid \vec{q}\right)} \quad \ldots(11)$$

- 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-relevant, 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 domain. 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. $P(R = 1 \mid \vec{q})$ 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 $P(R = 1 \mid \vec{q})$ and $P(R = 0 \mid \vec{q})$. Since a document is either relevant or non-relevant to a query, we must have that:

$$P(R=1 \mid \vec{x}, \vec{q}) + P(R=0 \mid \vec{x}, \vec{q}) = 1 \quad \ldots(12)$$

Chapter Ends...

❑❑❑