TABLE OF CONTENTS

Chapter - 1 Introduction to Information Retrieval (1 - 1) to (1 - 18)

1.1 History of IR ..... 1 - 2

1.2 Components of IR ..... 1 - 2

1.2.1 Process of Information Retrieval ..... 1 - 4

1.3 Data Retrieval and Information Retrieval ..... 1 - 5

1.3.1 Difference between Data Retrieval and Information Retrieval ..... 1 - 5

1.4 Data ..... 1 - 6

1.4.1 Unstructured Data ..... 1 - 8

1.4.2 Difference between Structured and Unstructured Data ..... 1 - 8

1.4.3 Semi-structured Text ..... 1 - 9

1.5 Inverted Index ..... 1 - 9

1.6 Boolean Queries ..... 1 - 12

1.7 Multiple Choice Questions with Answers ..... 1 - 15

Chapter - 2 Text Indexing, Storage and Compression (2 - 1) to (2 - 10)

2.1 Text Encoding ..... 2 - 2

2.1.1 Tokenization ..... 2 - 2

2.1.2 Stemming ..... 2 - 2

2.1.3 Stop Words ..... 2 - 3

2.1.4 Phrases ..... 2 - 4

2.2 Index Compression ..... 2 - 5

2.3 Zipf's Law ..... 2 - 5

2.4 Index Construction ..... 2 - 6

2.4.1 Dynamic Indexing ..... 2 - 8

2.4.2 Positional Indexes ..... 2 - 8

2.5 Multiple Choice Questions with Answers ..... 2 - 9

Scanned by TapScanner

Chapter - 3 Retrieval Models **(3 - 1) to (3 - 38)**

3.1 Information Retrieval Models ..... 3 - 2
3.1.1 Basic Concept ..... 3 - 2
3.1.2 Boolean Model ..... 3 - 2
3.1.3 Vector Model ..... 3 - 4
3.2 Term Weighting ..... 3 - 6
3.2.1 TF-IDF Weighting ..... 3 - 7
3.2.2 Luhn's Ideas ..... 3 - 8
3.2.3 Conflation Algorithm ..... 3 - 9
3.2.4 Cosine Similarity ..... 3 - 11
3.3 Inverted File ..... 3 - 11
3.3.1 Searching ..... 3 - 13
3.3.2 Construction ..... 3 - 14
3.4 Language Model based IR ..... 3 - 14
3.5 Probabilistic IR ..... 3 - 16
3.6 Latent Semantic Indexing ..... 3 - 18
3.7 Relevance Feedback and Query Expansion ..... 3 - 20
3.7.1 Rocchio Method ..... 3 - 22
3.7.2 Precision and Recall ..... 3 - 24
3.7.2.1 Interpolated Recall-Precision ..... 3 - 27
3.7.2.2 Mean Average Precision (MAP) ..... 3 - 29
3.7.3 Probability Relevance Feedback ..... 3 - 32
3.7.4 Pseudo Relevance Feedback ..... 3 - 32
3.7.5 Indirect Relevance Feedback ..... 3 - 34
3.8 Query Generation Probability ..... 3 - 34
3.9 Document Length Normalization ..... 3 - 36
3.10 Multiple Choice Questions with Answers ..... 3 - 36

Chapter - 4 Performance Evaluation **(4 - 1) to (4 - 18)**

4.1 Evaluating Search Engines ..... 4 - 2
4.1.1 TREC Collection ..... 4 - 3

Scanned by TapScanner

4.1.2 Benefit and Drawback of TREC . . . . . 4 - 8
4.2 User Happiness . . . . . 4 - 8
4.2.1 Search Engine Optimization . . . . . 4 - 9
4.2.2 Combining Searching and Browsing . . . . . 4 - 11
4.3 Creating Test Collections . . . . . 4 - 12
4.3.1 The Cranfield Collection . . . . . 4 - 12
4.3.2 The CACM and ISI Collection . . . . . 4 - 13
4.3.3 Kappa Measure for Inter - Judge Agreement . . . . . 4 - 14
4.4 Multiple Choice Questions with Answers . . . . . 4 - 15

Chapter - 5 Text Categorization and Filtering (5 - 1) to (5 - 16)

5.1 Introduction to Text Classification . . . . . 5 - 2
5.1.1 Machine Learning . . . . . 5 - 2
5.1.2 Text Classification Problem . . . . . 5 - 4
5.1.3 Text Classification Algorithm . . . . . 5 - 4
5.2 Naive Bayes Models . . . . . 5 - 4
5.2.1 Spam Filtering . . . . . 5 - 5
5.3 k-Nearest Neighbors . . . . . 5 - 6
5.4 Support Vector Machine Classifiers . . . . . 5 - 9
5.4.1 Kernel Functions . . . . . 5 - 11
5.5 Boosting . . . . . 5 - 12
5.6 Multiple Choice Questions with Answers . . . . . 5 - 14

Chapter - 6 Text Clustering (6 - 1) to (6 - 14)

6.1 Clustering . . . . . 6 - 2
6.1.1 Clustering Versus Classification . . . . . 6 - 4
6.2 Partitioning Methods . . . . . 6 - 4
6.2.1 K-Mean Clustering . . . . . 6 - 5
6.2.2 K-Medoids . . . . . 6 - 6
6.2.3 Silhouette . . . . . 6 - 7
6.3 Mixture of Gaussians Model . . . . . 6 - 8

Scanned by TapScanner

6.4 Hierarchical Agglomerative Clustering ..... 6 - 9
6.4.1 Agglomerative Clustering ..... 6 - 9
6.4.2 Dendrogram ..... 6 - 11
6.4.3 Difference between Hierarchical Clustering and K-Mean Clustering ..... 6 - 12
6.5 Clustering Terms using Documents ..... 6 - 12
6.6 Multiple Choice Questions with Answers ..... 6 - 13

Chapter - 7 Advanced Topics **(7 - 1) to (7 - 10)**

7.1 Summarization ..... 7 - 2
7.2 Topic Detection and Tracking ..... 7 - 2
7.3 Personalization ..... 7 - 4
7.4 Question Answering ..... 7 - 6
7.5 Cross Language Information Retrieval ..... 7 - 6

Chapter - 8 Web Information Retrieval **(8 - 1) to (8 - 30)**

8.1 The Web ..... 8 - 2
8.1.1 Characteristics ..... 8 - 3
8.1.2 Modeling the Web ..... 8 - 3
8.1.3 Link Analysis ..... 8 - 5
8.2 Hypertext ..... 8 - 6
8.3 Web Crawling ..... 8 - 6
8.3.1 Web Crawler Architecture ..... 8 - 7
8.3.2 Taxonomy of Crawler ..... 8 - 9
8.4 Search Engines ..... 8 - 9
8.4.1 Cluster based Architecture ..... 8 - 9
8.4.2 Distributed Architecture ..... 8 - 12
8.5 Ranking ..... 8 - 15
8.5.1 Link based Ranking ..... 8 - 15
8.5.2 Simple Ranking Functions ..... 8 - 17
8.5.3 Learning to Rank ..... 8 - 18
8.5.4 Evaluations ..... 8 - 18

Scanned by TapScanner

8.6 Link Analysis ..... 8 - 19
8.6.1 Finding Needle in the Haystack ..... 8 - 20
8.6.2 Hubs and Authorities ..... 8 - 21
8.7 PageRank & HITS ..... 8 - 23
8.7.1 Page Rank Algorithm ..... 8 - 23
8.7.2 HITS ..... 8 - 26
8.7.3 Difference between HITS and PageRank. ..... 8 - 28
8.8 Multiple Choice Questions with Answers ..... 8 - 29

Chapter - 9 Retrieving Structured Documents (9 - 1) to (9 - 18)

9.1 XML Retrieval ..... 9 - 2
9.1.1 Types of Queries ..... 9 - 2
9.1.2 Patterns Matching ..... 9 - 3
9.1.3 Structural Queries ..... 9 - 3
9.2 Semantic Web ..... 9 - 4
9.2.1 Development of Semantic Web ..... 9 - 5
9.2.2 Introduction Knowledge Representation on the Semantic Web ..... 9 - 5
9.2.3 Ontology and their Role in the Semantic Web ..... 9 - 7
9.2.4 Ontology Languages for the Semantic Web ..... 9 - 9
9.2.4.1 Resource Description Framework (RDF) and RDF Schema ..... 9 - 9
9.2.4.2 Web Ontology Language (OWL) ..... 9 - 15

Scanned by TapScanner

1 Introduction to Information Retrieval

Syllabus

The nature of unstructured and semi - structured text. Inverted index and Boolean queries.

Contents

1.1 History of IR

1.2 Components of IR

1.3 Data Retrieval and Information Retrieval

1.4 Data

1.5 Inverted index

1.6 Boolean Queries

1.7 Multiple Choice Questions

Scanned by TapScanner

1.1 History of IR

- Information Retrieval (IR) is about the process of providing answers to client's information needs. It is thus concerned with the collection, representation, storage, organization, accessing, manipulation and display, of the information items necessary to satisfying those needs.
- Definition : Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections.
- There is a huge quantity of text, audio, video and other documents available on the Internet, on about any subject. Users need to be able to find relevant information to satisfy their particular information needs.
- There are two ways of searching for information : To use a search engines or to browse directories organized by categories.
- IR is the task of representing, storing, organizing, and offering access to information items. IR is different from data retrieval, which is about finding precise data in databases with a given structure.
- In IR systems, the information is not structured. It is contained in free form in text (web pages or other documents) or in multimedia content.
- The first IR systems implemented in 1970's were designed to work with small collections of text. Some of these techniques are now used in search engines.
- The information retrieval techniques focusing on the challenges faced by search engine.
  1. One particular challenge is the large scale, given by the huge number of web-pages available on the Internet.
  2. The ambiguity of the natural language (English or other languages) that makes it difficult to have perfect matches between documents and user queries.
- Information retrieval is never an easy task. The problem with IR is that document representation, either by index terms or texts cannot satisfy user need representation, which is dynamic and complicated.
- Traditional IR systems are designed to support only one type of information-seeking strategy that users engage in : Query formulation.

1.2 Components of IR

- An information retrieval system is an information system, which is used to store items of information that need to be processed, searched, retrieved, and disseminated to various user populations.
- Information retrieval is the process of searching some collection of documents, in order to identify those documents which deal with a particular subject. Any

Scanned by TapScanner

system that is designed to facilitate this literature searching may legitimately be called an information retrieval system.

- Conceptually, IR is the study of finding needed information. It helps users to find information that matches their information needs. Historically, IR is about document retrieval, emphasizing document as the basic unit.
- Information retrieval locates relevant documents, on the basis of user input such as keywords or example documents, for example : Find documents containing the words "database systems".
- Fig. 1.2.1 shows information retrieval system block diagram. It consists of three components : **Input, processor and output.**

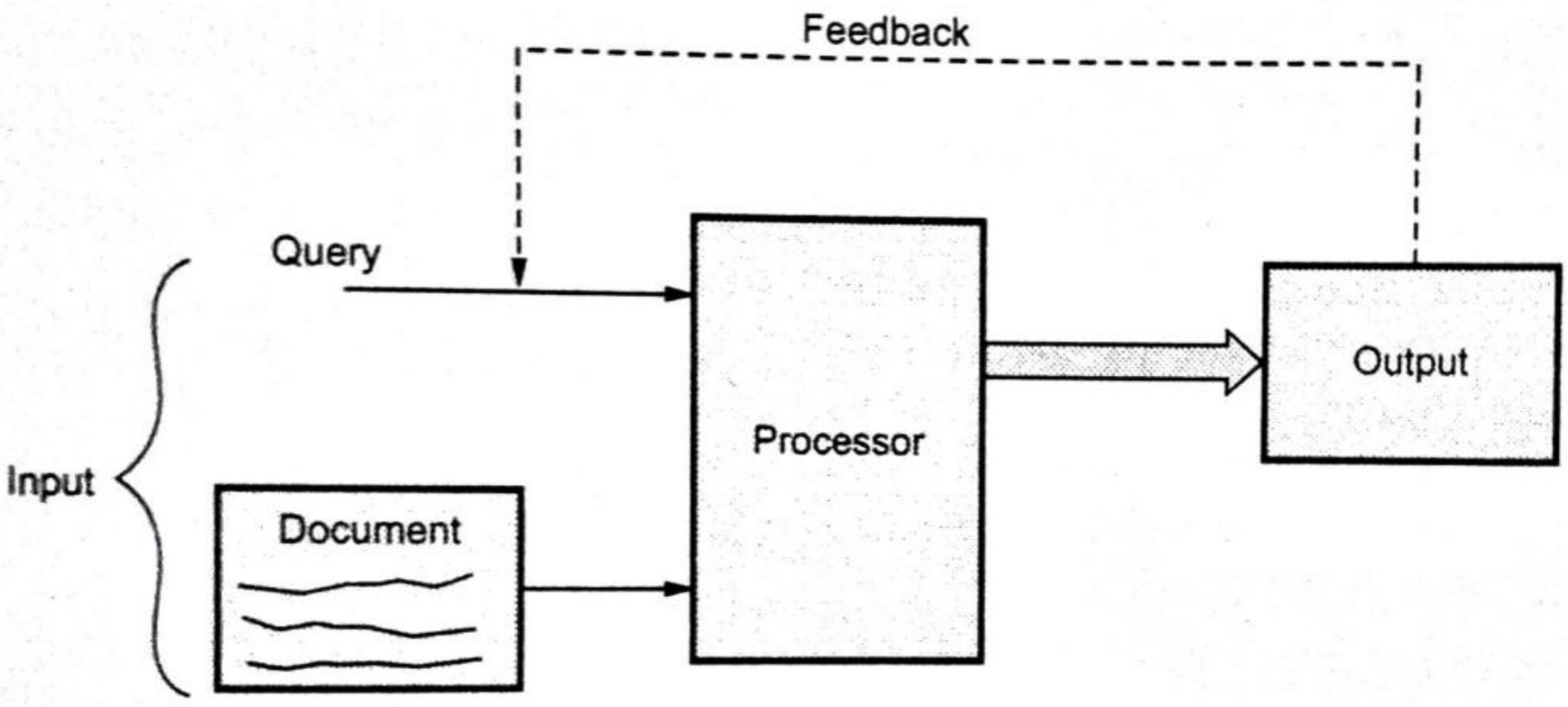


Fig. 1.2.1 IR block diagram

a) **Input :** Store only a representation of the document or query which means that the text of a document is lost once it has been processed for the purpose of generating its representation.

b) A **document representative** could be a list of extracted words considered to be significant.

c) **Processor :** Involve in performing actual retrieval function, executing the search strategy in response to a query.

d) **Feedback :** Improving the subsequent run after sample retrieval.

e) **Output :** A set of document numbers.

- Information retrieval locates relevant documents, on the basis of user input such as keywords or example documents.
- The computer-based retrieval systems store only a representation of the document or query which means that the text of a document is lost once it has been processed for the purpose of generating its representation.

Scanned by TapScanner

- The process may involve structuring the information, such as classifying it. It will also involve performing the actual retrieval function that is executing the search strategy in response to a query.
- Text document is the output of information retrieval system. Web search engines are the most familiar example of IR systems.

1.2.1 Process of Information Retrieval

- Information retrieval is often a continuous process during which you will consider, reconsider and refine your research problem, use various different information resources, information retrieval techniques and library services and evaluate the information you find.
- Fig. 1.2.2 shows that the stages follow each other during the process, but in reality they are often active simultaneously and you usually will repeat some stages during the same information retrieval process.
- The different stages of the information retrieval process are :

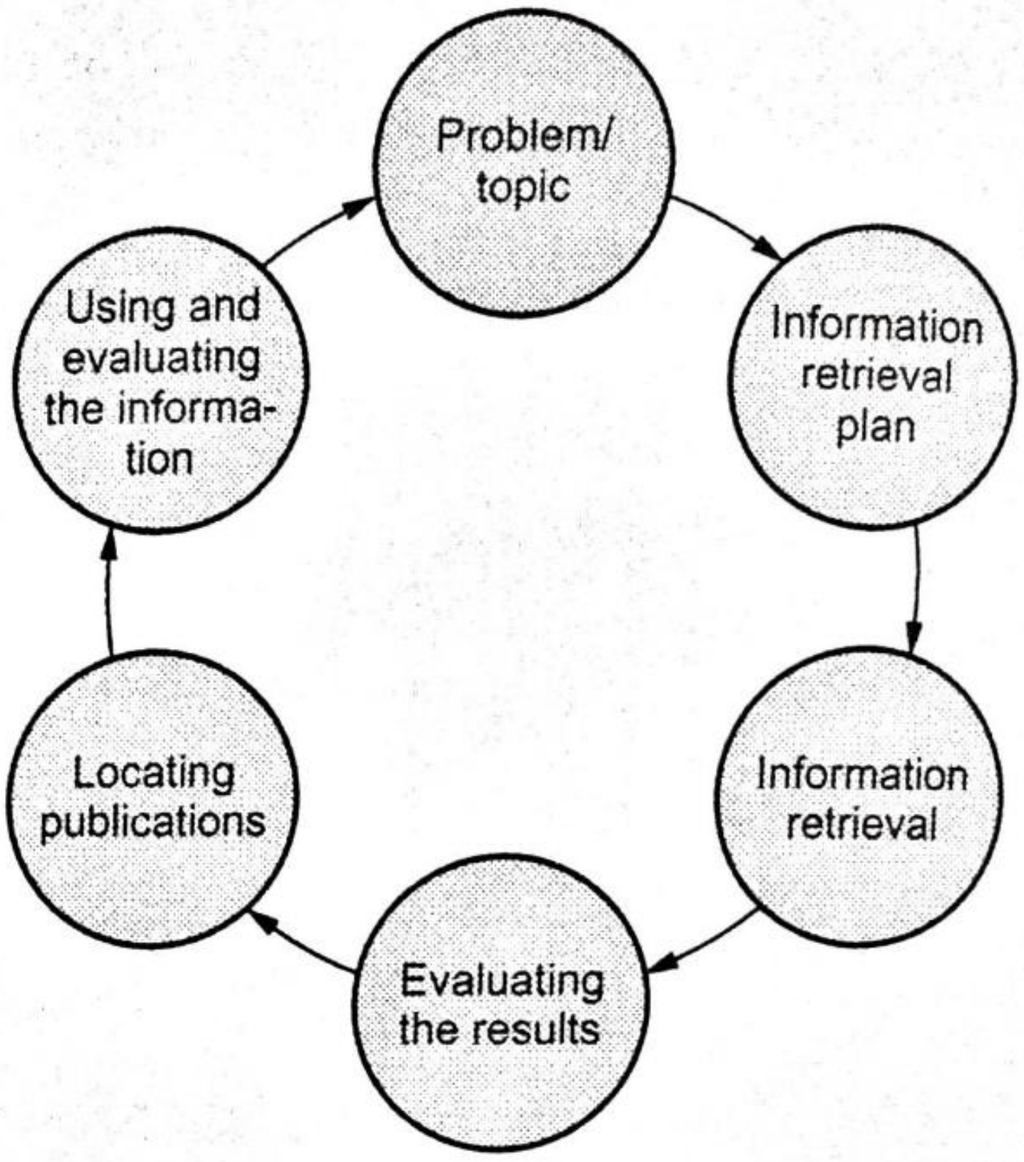


Fig. 1.2.2

1. Problem / Topic : An information need occurs when more information is required to solve a problem
2. Information retrieval plan : Define your information need and choose your information resources, retrieval techniques and search terms
3. Information retrieval : Perform your planned information retrieval (information retrieval techniques)
4. Evaluating the results : Evaluate the results of your information retrieval (number and relevance of search results)
5. Locating publications : Find out where and how the required publication, e.g. article, can be acquired
6. Using and evaluating the information : Evaluate the final results of the process (critical and ethical evaluation of the information and information resources)

Scanned by TapScanner

1.3 Data Retrieval and Information Retrieval

- An information retrieval system is software that has the features and functions required to manipulate "information" items versus a DBMS that is optimized to handle "structured" data.
- Information retrieval and Data Retrieval (DR) are often viewed as two mutually exclusive means to perform different tasks, IR being used for finding relevant documents among a collection of unstructured/semi-structured documents.
- Data retrieval being used for finding exact matches using stringent queries on structured data, often in a Relational Database Management System (RDBMS).
- IR is used for assessing human interests, i.e., IR selects and ranks documents based on the likelihood of relevance to the user's needs. DR is different; answers to users' queries are exact matches which do not impose any ranking.
- Data retrieval involves the selection of a fixed set of data based on a well-defined query. Information retrieval involves the retrieval of documents of natural language.
- IR systems don't support transactional updates whereas database systems support structured data, with schemas that define the data organization. IR systems deal with some querying issues not generally addressed by database systems and approximate searching by keywords.

1.3.1 Difference between Data Retrieval and Information Retrieval

Parameters	Data retrieval	Information retrieval
Example	Data base query	WWW search
Matching	Exact	Partial match Best match
Inference	Deduction	Induction
Model	Deterministic	Probabilistic
Query language	Artificial	Natural
Query specification	Complete	Incomplete
Items wanted	Matching	Relevant
Error response	Sensitive	Insensitive
Classification	Monotonic	Polytechnic

Scanned by TapScanner

1.4 Data

- Data is collection of facts and figures which relay something specific, but which are not organized in any way. It can be numbers, words, measurements, observations or even just descriptions of things. We can say, data is raw material in the production of information.
- **Examples of data :** Printed paper, bank account passbook, student attendance, salary sheet etc.
- Data creation is limited because of lack of new technology. After start of using computer, data can be converted into more convenient forms. It starts of using email, e-book, digital audio and video, images etc.
- Data is raw. It simply exists and has no significance beyond its existence. It can exist in any form, usable or not. It does not have meaning of itself. In computer parlance, a spreadsheet generally starts out by holding data.
- Facts and figures which relay something specific, but which are not organized in any way and which provide no further information regarding patterns, context, etc.
- Data represents a fact or statement of event without relation to other things. Ex : It is raining. Computer stores data in the form of 0 and 1. This is called digital data. These data is stored in the computer in the form of 0 and 1. Data in this form is called digital data. It is shown in Fig. 1.4.1.

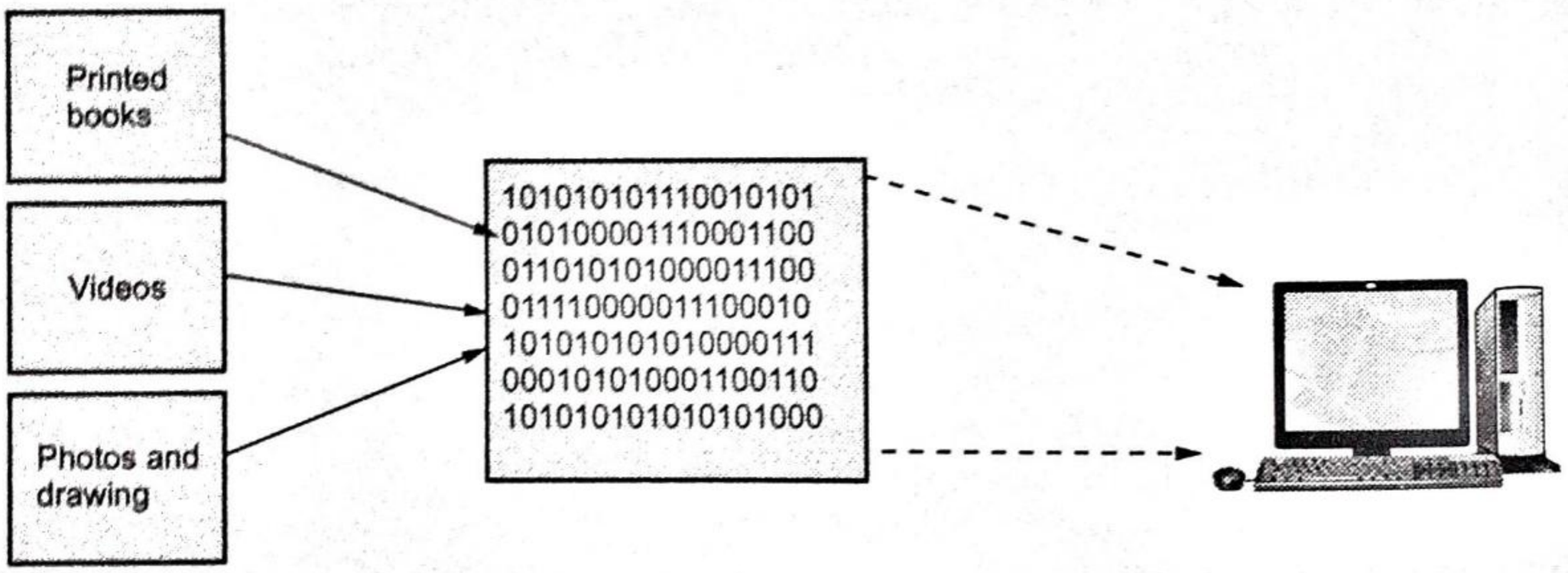


Fig. 1.4.1 : Digital data

- Data is raw. It has not been shaped, processed or interpreted. Once data has been processed and turned into information. Computers need data. Humans need information. Data is a building block. Information gives meaning and context.

Scanned by TapScanner

- Information systems are combinations of hardware, software, and telecommunications networks that people build and use to collect, create, and distribute useful data, typically in organizational settings.
- Company offers their products to customers for making money. These products can be goods or services. Large number of different types of data is accumulated in the organization. It contains data of products, customer data, employee's data, data of the delivery of products, and data of other sources.
- These data must be stored, managed and processed. For performing these operations on data, information system can play an important role. Because this data is raw data. There is no unique format.
- Because of new technology, generation of new data and sharing of data has increased exponentially. The factors that affects the growth of digital data :
  1. Cost of digital storage device is decreases.
  2. Data processing capabilities is also increases.
  3. Faster access of data.

Types of Data

- Data are of two types : structured and unstructured

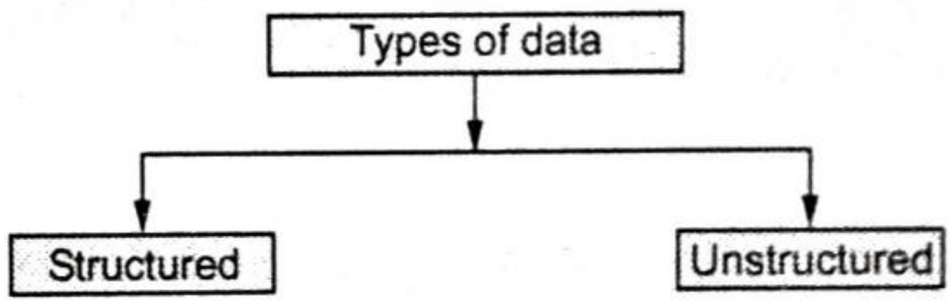


Fig. 1.4.2

Structured Data :

- Structured data is arranged in rows and column format. It helps for application to retrieve and process data easily. Database management system is used for storing structured data.
- The term structured data refers to data that is identifiable because it is organized in a structure. The most common form of structured data or records is a database where specific information is stored based on a methodology of columns and rows.
- Structured data is also searchable by data type within content. Structured data is understood by computers and is also efficiently organized for human readers.

Scanned by TapScanner

1.4.1 Unstructured Data

- Unstructured data is data that does not follow a specified format. Row and columns are not used for unstructured data. Therefore it is difficult to retrieve required information. Unstructured data has no identifiable structure.
- The unstructured data can be in the form of text : (documents, email messages, customer feedbacks) , audio, video, images. Email is an example of unstructured data.
- Even today in most of the organizations more than 80 % of the data are in unstructured form. This carries lots of information. But extracting information from these various sources is a very big challenge.
- Characteristics of unstructured data,
  1. There is no a structural restriction or binding for the data.
  2. Data can be of any type.
  3. Unstructured data does not follow any structural rules.
  4. There are no predefined formats, restriction or sequence for unstructured data.
  5. Since there is no structural binding for unstructured data, it is unpredictable in nature.
- Examples of machine generated unstructured data :
  1. Satellite images : This includes weather data or the data that the government captures in its satellite surveillance imagery.
  2. Scientific data : This includes atmospheric data, and high energy physics.
  3. Photographs and video : This includes security, surveillance, and traffic video.

1.4.2 Difference between Structured and Unstructured Data

Parameters	Structured Data	Unstructured Data
Representation	It is in discrete form. i. e stored in row and column format.	Unstructured data is data that does not follow a specified format
Metadata	Syntax	Semantics
Storage	Database management system	Unmanaged file structure
Standard	SQL, ADO.net, ODBC	Open XML, SMTO, SMS
Tools for integration	ETL	Batch processing or manual data entry

Scanned by TapScanner

Characteristics	With a structured document, certain information always appears in the same location on the page.	In unstructured document, information can appear in unexpected places on the document.
Used by Organizations	Low volume operations	High volume operations

1.4.3 Semi-structured Text

- Semi-structured data as a type of data that contains semantic tags, but does not conform to the structure associated with typical relational databases.
- While semi-structured entities belong in the same class, they may have different attributes. Examples include email, XML and other markup languages.
- A good example of semi-structured data is HTML code, which doesn't restrict the amount of information you want to collect in a document, but still enforces hierarchy via semantic elements.

1.5 Inverted Index

- An index into a set of texts of the words in the texts. The index is accessed by some search method. Each index entry gives the word and a list of texts, possibly with locations within the text, where the word occurs.
- Inverted index : A word-oriented mechanism for indexing a text collection to speed up the searching task.
- The inverted index is a database index storing a mapping from content, such as words or numbers, to its locations in a database, or in a document or a set of documents. The purpose of an inverted index is to allow a fast full-text search.
- The inverted index structure is composed of two elements : the vocabulary and the occurrences.
- The vocabulary is the set of all different words in the text. For each word in the vocabulary the index stores the documents which contain that word.
- Let us consider following example with two documents :
  a) Document#1 with text "Recipe of pasta with sauce pesto"
  b) Document#2 with text "Recipe of delicious carbonara pasta"
- To build the inverted index, we have to split those sentences into tokens, in our case, the tokens are words. After this, tokens will be saved with links to documents, in some cases, before saving tokens can be transformed.

Scanned by TapScanner

- The inverted index for those sentences will look like this simplified schema :

Token	Document#1	Document#2
Carbonara		+
delicious		+
of	+	+
pasta	+	+
pesto	+	
recipe	+	+
sauce	+	
with	+	

- 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.
- In this inverted index, we can find a link to the document by token(a word from the text). Let's look at a few examples of search queries :
  a) The search query "pasta recipe" will be divided into two tokens ("pasta" and "recipe"), then it will be matched with the inverted index and will return both documents with the same score (the score is a similarity of a document and a search phrase; we will review the score later in this article).
  b) The search query "carbonara pasta" will give you both documents as well, but document#2 will have a bigger score then document#1. Because document#2 has both tokens (and "carbonara" and "pasta"), but document#1 has only one("pasta") token.
  c) The search query "delicious carbonara" will give you only document#2, because this document has both tokens, and document#1 has none of them.
- Sometimes we can transform tokens before saving and searching. The most popular kinds of transformation are :
  a) Dropping of the stop words. Stop words are the most popular words in the language or in your dataset, which don't have any semantic weight. For example, in English, it can be the following words "of", "the", "for", etcetera.
  b) Lemmatization. Lemmatization is a process of changing a word to a dictionary form of the word, "running" => "run", "walks" => "walk", "thought" =>"think"
  c) Stemming. Stemming is a process of transforming a word into root form by cutting the ending of the word. This is similar to Lemmatization, but can not handle cases with irregular verbs, but can handle words which are not in the dictionary.

Scanned by TapScanner

Full- Inverted Index

- Term-document matrix : The simplest way to represent the documents that contain each word of the vocabulary.
- Let us consider following four documents.

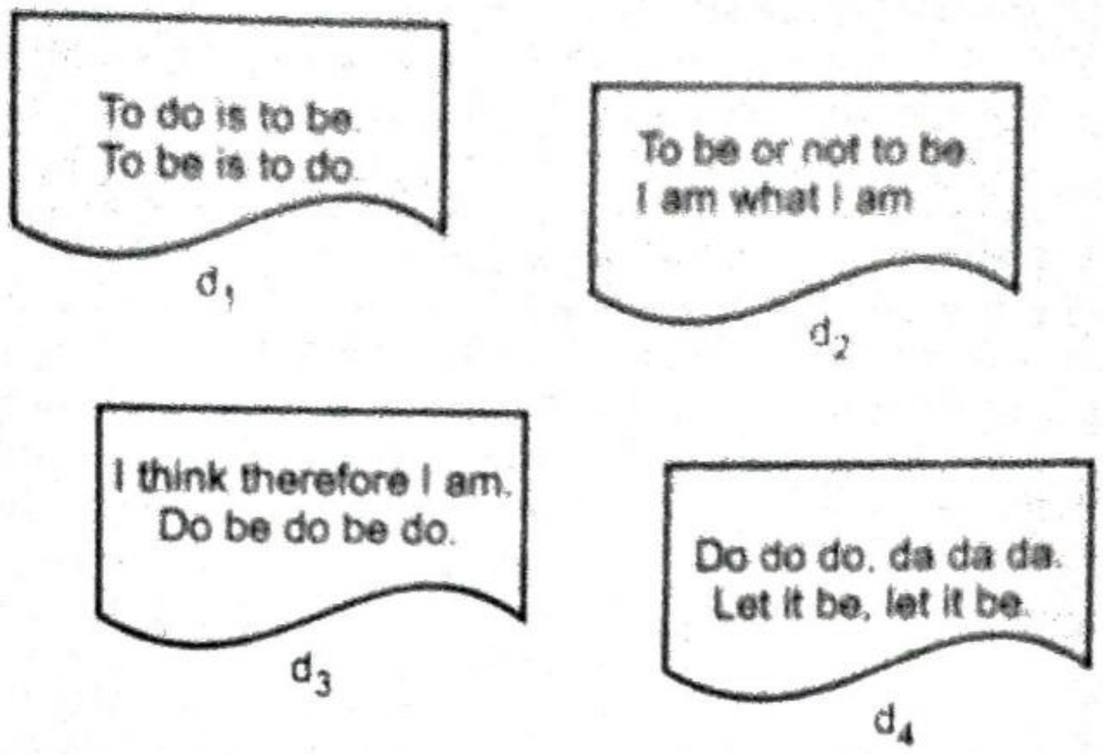


Fig. 1.5.1

- Following Fig. 1.5.2 shows vocabulary with inverted list.

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. 1.5.2 Vocabulary with inverted list

Scanned by TapScanner

- 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.**
- 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)

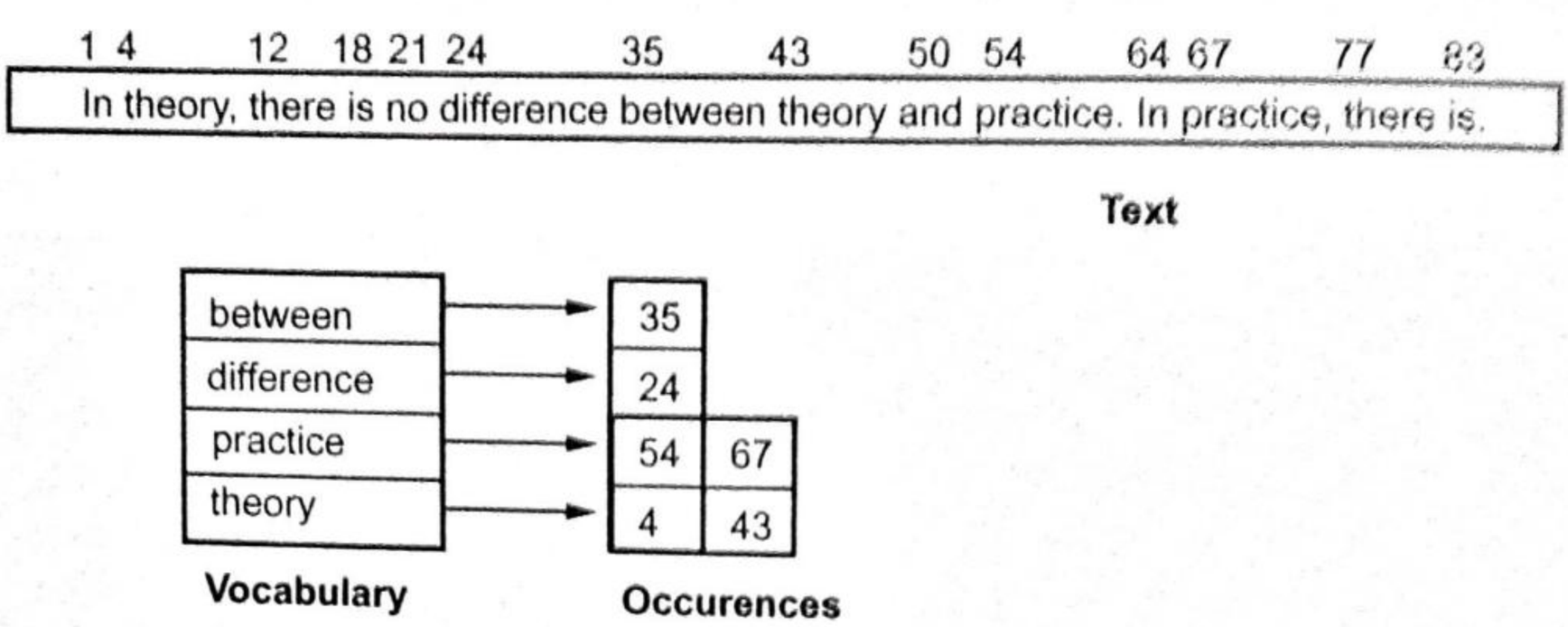


Fig. 1.5.3

- The space required for the vocabulary is rather small. 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.
- Document-addressing indexes are smaller, because only one occurrence per file must be recorded, for a given word. 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 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.

1.6 Boolean Queries

- A query that matches documents matching boolean combinations of other queries.
- Boolean searches are named after the British born Mathematician George Boole. Boolean logic establishes the relationship between keywords in a search. Boolean logic has three operators : AND, OR , NOT.
- A Boolean search strategy retrieves those documents which are 'true' for the query. This formulation only makes sense if the queries are expressed in terms of index terms or keywords and combined by the usual logical connectives AND, OR and NOT.

Scanned by TapScanner

- For example, if the query Q = (K_1 AND K_2) OR (K_3 AND (NOT K_4)) then the Boolean search will retrieve all documents indexed by K_1 and K_2, as well as all documents indexed by K_3 which are not indexed by K_4.

1. Search keywords Iceland AND Whales :

This search will give you all pages with both the keyword Iceland and Whales contained in the page or metadata. This is the basic search that most people use on search engines like Google. This search will find any page that has the words **Iceland** and **Whales** in the page. The search engines usually look for words within the body of the page or in the Metadata in the head of the page.

2. Search keywords Iceland OR Whales :

This search will give you all pages with either the keyword Iceland or Whales contained in the page or metadata. This search will produce many hits as it looks for pages with **either of the** two keywords within the body of the page or in the Metadata in the head. The results will include :

a. Pages with "Whales"

b. Pages with "Iceland"

c. Pages containing both "Whales" and "Iceland"

This style of Boolean search is useful when you have related topics or where there are several names for the same topics. The OR operator is used when the thing you are looking for has more than one legitimate name. For example : "Standards of Learning" is also called "SOL".

3. Search keywords Iceland NOT Whales :

This search will give you all pages with the keyword Iceland but not the keyword Whales contained in the page or metadata. The NOT search is a powerful filtering tool that allows the searcher to eliminate pages quickly and easily. It is particularly useful when combined with other search terms.

- On some systems, the Boolean search allow the user to narrow or broaden the search by giving the user access to a structured dictionary which, for any given keyword, stores related keywords which may be more general or more precise.
- Consider example of tree structure. Following Fig. 1.6.1 shows a set of hierarchically related keywords.
- 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. If one has an interactive system the search can easily be reformulated using some of these related terms.

Scanned by TapScanner

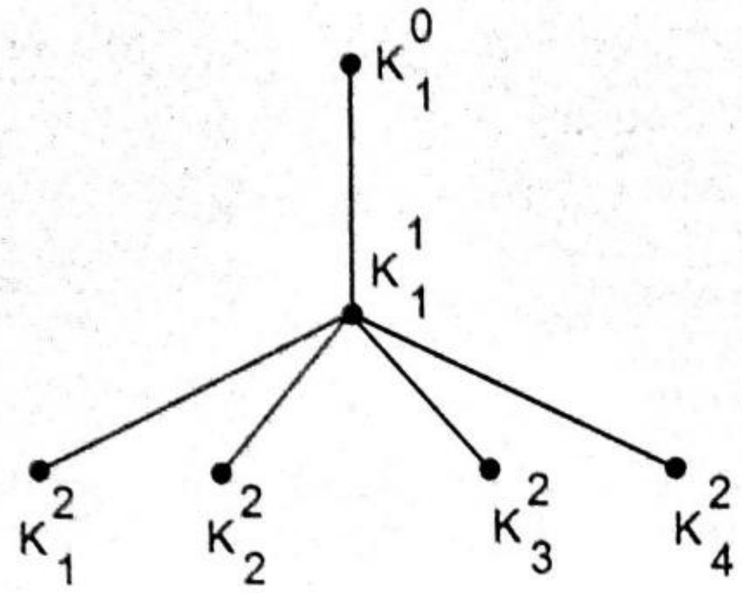


Fig. 1.6.1 Set of hierarchically related keywords

- Boolean search is implemented using inverted file. An inverted file is a list of keywords and identifiers of the documents in which they occur. List for each keyword is stored in the vocabulary and in each list put the addresses or numbers of the documents containing that particular word.
- To satisfy query, the set operations is performed on the K_i - lists corresponding to the logical connectives. 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 $Q = (K_1 \text{ AND } K_2) \text{ OR } (K_3 \text{ AND } (\text{NOT } K_4))$

  then to satisfy the $(K_1 \text{ AND } K_2)$ part we intersect the K_1 and K_2 lists, to satisfy the $(K_3 \text{ AND } (\text{NOT } K_4))$ part, we subtract the K_4 list from the K_3 list.
- The OR is satisfied by now taking the union of the two sets of documents obtained for the parts. The result is the set $\{D_1, D_2, D_3\}$ which satisfies the query and each document in it is 'true' for the query.
- Full Boolean search is allowed for slight modification by using only AND logic. It consider the ac actual number of terms the query has in common with a document. This number has become known as the **co-ordination level.** This is also called **simple matching.** The documents are partially ranked by the co-ordination levels.
- For example : Query $Q = K_1 \text{ AND } K_2 \text{ AND } K_3$, we obtain the following ranking :

  Co-ordination level

  3 D_1, D_2

  2 D_3

  1 D_4

Scanned by TapScanner

- A general problem with Boolean search is that using AND operators tends to produce high precision but low recall searches, while using OR operators gives low precision but high recall searches and it is difficult or impossible to find a satisfactory middle ground.

Advantages of Boolean Search :

1. Simple queries are easy to understand.
2. Relatively easy to implement.

Disadvantages of Boolean Search :

1. Difficult to specify what is wanted.
2. Too much returned or too little.
3. Ordering is not well determined.

1.7 Multiple Choice Questions

Q.1 **_____ retrieval deals with the representation, storage, organization of and access to information items such as documents, web pages, online catalogs, structured and semi-structured records and multimedia objects.**

a Data
b Audio
c Video
d Information

Q.2 **_____ of a retrieval system has to translate his information need into a query in the language provided by the system.**

a The manager
b The user
c The designer
d The administrator

Q.3 **_____ is usually provided by most modern information retrieval systems.**

a Information and knowledge retrieval
b Information or knowledge retrieval
c Information and data retrieval
d Information or data retrieval

Q.4 **_____ is an iterative process of formulating a conceptual from a large collection of information.**

a Sensemaking
b Data collection
d Information collection
d All of these

Scanned by TapScanner

Q.5 A web page is an _____ document that is stored on a web server.

a XML
b HTML
c XSL
d Java

Q.6 A _____ is a hypertext link to a page on a website other than its homepage.

a Hyper link
b deep link
c URL
d HTML

Q.7 Which methods are used in search engine for visualization concept ?

a Visualizing Boolean syntax
b Visualizing query terms within retrieval results
c Visualizing relationships among words and documents
d All of these

Q.8 URL Stands for _____.

a Uniform Ravar Location
b Uniform Resource Locator
c Uni Resource Locate
d Uniform Reverse Locator

Q.9 Which of the following is a search engine ?

a Google
b Yahoo!
c Bing
d All of these

Q.10 In IR systems, the information is _____.

a structured
b semi - structured
c not structured
d none

Q.11 A search engine is a program to search _____.

a for information
b web pages for information using specified search terms
c web pages
d web pages for specified index terms

Scanned by TapScanner

Q.12 **Early IR systems were _____ systems which allowed users to specify their information need using a complex combination of Boolean ANDs, ORs and NOTs.**

a Boolean
b vector
c logical
d All of these

Q.13 **Information Retrieval Systems is characterized by _____ data format.**

a structured
b semi - structured
c unstructured
d All of these

Q.14 **_____ are a set of electronic resources and associated technical capabilities for creating, searching, and using information.**

a Analog libraries
b Digital libraries
c Digital information
d Digital data

Q.15 **Which of the following is NOT components of IR block diagram ?**

a Input
b Processor
c Feedback
d Information

Q.16 **Web browser is a software program that interprets and displays the contents of _____ web pages.**

a XML
b HTML
c static
d dynamic

Q.17 **_____ diagram is a diagram that visually displays all the possible logical relationships between collections of sets.**

a Text
b Information
c Binary
d Venn

Q.18 **A\An _____ is an index into a set of documents of the words in the documents.**

a inverted index
b decision tree
c confusion matrix
d all of these

Scanned by TapScanner

Answer Keys for Multiple Choice Questions

Q.1	d	Q.2	b	Q.3	c	Q.4	a
Q.5	b	Q.6	b	Q.7	d	Q.8	b
Q.9	d	Q.10	c	Q.11	b	Q.12	a
Q.13	c	Q.14	b	Q.15	d	Q.16	b
Q.17	d	Q.18	a				

❑❑❑

Scanned by TapScanner

2 Text Indexing, Storage and Compression

Syllabus

Text encoding: tokenization, stemming, stop words, phrases, index optimization. Index compression: lexicon compression and postings lists compression. Gap encoding, gamma codes, Zipf's Law. Index construction. Postings size estimation, merge sort, dynamic indexing, positional indexes, n-gram indexes, real-world issues.

Contents

2.1 *Text Encoding*

2.2 *Index Compression*

2.3 *Zipf's Law*

2.4 *Index Construction*

2.5 *Multiple Choice Questions*

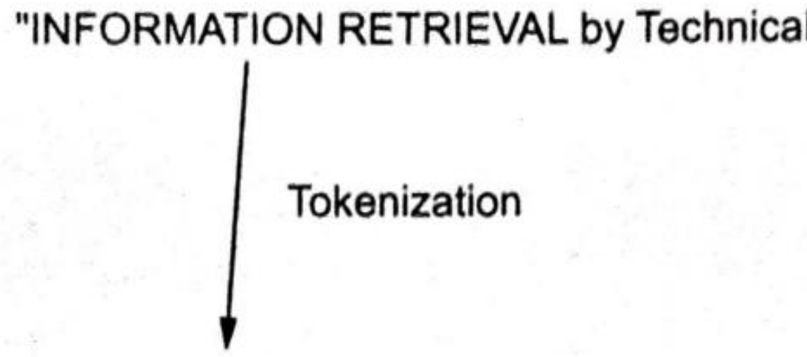
Scanned by TapScanner

2.1 Text Encoding

- A character encoding provides a key to unlock the code. It is a set of mappings between the bytes in the computer and the characters in the character set. Without the key, the data looks like garbage.

2.1.1 Tokenization

- Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. The simplest form of analysis is to reduce different word forms into tokens.
- Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords.
- Tokenization can be broadly classified into 3 types : Word, character, and subword (n-gram characters) tokenization.
- These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.



- Tokenization can be done to either separate words or sentences. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing.
- Tokens are the building blocks of Natural Language, the most common way of processing the raw text happens at the token level.
- The major question of the tokenization phase is what are the correct tokens to use ? You chop on whitespace and throw away punctuation characters.
- Types of tokenization are white space, dictionary based, rule based, penn tree, spacy, subword etc.

2.1.2 Stemming

- Stemming is the process for reducing inflected words to their stem, base or root form, generally a written word form. Stemming allows a query term such as "orienteering" to match an occurrence of "orienteers", or "runs" to match "running".

Scanned by TapScanner

- For example, "orienteering" and "orienteers" might reduce to the root form "orienteer"; "runs" and "running" might reduce to "run".
- In an IR system a stemmer may be applied at both indexing time and query time. During indexing each token is passed through the stemmer and the resulting root form is indexed.
- At query time, the query terms are passed through the same stemmer and matched against the index terms. Thus the query term "runs" would match an occurrence of "running" by way of their common root form "run".
- Stemming is the particular case of tokenization which reduces inflected forms to a single base form or stem. Stemming algorithms are basic string - handling algorithms, which depend on rules which identify affixes that can be stripped
- The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance :

  am, are, is ⇒ be

  car, cars, car's, cars' ⇒ car
- The result of this mapping of text will be something like : The boy's cars are different colors the boy car be differ color

2.1.3 Stop Words

- The removal of high frequency words, 'stop' words or 'fluff' words is one way of implementing Luhn's upper cut-off. This is normally done by comparing the input text with a 'stop list' of words which are to be removed.
- Sometimes, some extremely common words that would appear to be of little value in helping select documents matching a user need are excluded from stop words the vocabulary entirely. These words are called stop words.
- Function words are words that have no well - defined meanings in and of themselves. Function words are usually among the most frequently occurring words in any language. At query time these stopwords are stripped from the query, and retrieval takes place on the basis of the remaining terms alone.
- Examples of a few stop words in English are "the", "a", "an", "so", "what".
- Removal of stop words definitely reduces the dataset size and thus reduces the training time due to the fewer number of tokens involved in the training.
- The phrase query "Capital of the India," which contains two stop words, is more precise than Capital AND "India".

Scanned by TapScanner

- Stop word list is as follows :

  a an and are as at be by for from has he in

  is it its of on that the to was were will with
- With stopwords present in the index, the IR system can make a decision on a query - by - query basis. Ranking methods in a modern commercial search engine will incorporate many ranking features, including features based on term frequency and proximity.
- For features that do not consider proximity between query terms, stopwords may be elminated. For features that do consider proximity between query terms, particularly to match their occurrence in phrases, it may be appropriate to retain stopwords.
- Good compression techniques means the space for including stopwords in a system is very small
- Good query optimization techniques mean you pay little at query time for including stopwords.

2.1.4 Phrases

- Many complex or technical concepts and many organization and product names are multiword compounds or phrases. An information retrieval system uses phrases to index, retrieve, organize and describe documents.
- Phrases are identified that predict the presence of other phrases in documents. Documents are the indexed according to their included phrases.
- Most recent search engines support a double quotes syntax ("technical publications") for phrase queries, which has proven to be very easily understood and successfully used by users.
- One approach to handling phrases is to consider every pair of consecutive terms in a document as a phrase. For example the text ***Friends, Romans, Countrymen*** would generate the biwords :

  friends romans

  romans countrymen
- Each of these biwords is now a dictionary term. Two - word phrase query - processing is now immediate.
- Biword indexes are not the standard solution but can be part of a compound strategy.

Scanned by TapScanner

2.2 Index Compression

- Uncompressed indexes are large. It might be useful for some modern devices to support information retrieval techniques that would not be able to do with uncompressed indexes.
- Types of Compression :
  1. Lossy : Compression that involves the removal of data.
  2. Loseless : Compression that involves no removal of data.
- Goal is to reduce the size of the index without reducing the precision such that a human can't tell the difference between a pruned index and non-pruned index.
- The index is the data structure that stores the information handled in the retrieval process. Two different approaches are proposed for index compression, namely document reordering and static index pruning.
- Compression needs a probability distribution for predicting the next symbol to be coded. The process of estimating the probability distribution is known as modelling, whereas the process of converting the symbols into bits (codewords) is known as coding.
- text compression methods are normally classified into :
  1. Symbolwise methods
  2. Dictionary methods
- Symbolwise methods work one symbol at a time, estimating the probability of every symbol, and usually assigning shorter code words to the most frequent symbols. Those symbols can be characters, words, or sequences of n characters/words. The size of the context used to encode those symbols decides the order of the encoder.
- Dictionary methods replace words and other fragments of text with an index to an entry in a dictionary. They use simple representations to code references to entries in the dictionary. They obtain good compression ratios by conflating several symbols into a single output codeword.
- Symbolwise methods are based on Huffman or arithmetic coding. The most successful dictionary methods are based on Ziv-Lempel coding

2.3 Zipf's Law

- Zipf's law is a law about the frequency distribution of words in a language.
- Let r be the rank of word, Prob(r) be the probability of a word at rank r. We do not consider the names of the words, we consider only about their ranks and frequencies.

Scanned by TapScanner

- By definition :

  Prob(r) = freq(r) / N

Where freq(r) = The number of times the word at rank r appears in the collection

N = Total number of words in the collection.

- Then Zipf's law states that

  r * Prob(r) = A,

where A is a constant which should empirically be determined from the data

2.4 Index Construction

- Many design decisions in information retrieval are based on the characteristics of hardware.
- Access to data in memory is much faster than access to data on disk.
- Disk seeks : No data is transferred from disk while the disk head is being positioned. Therefore, transferring one large chunk of data from disk to memory is faster than transferring many small chunks.
- Disk I/O is block-based : Reading and writing of entire blocks. Typical block sizes are 8 kB to 256 kB.
- Servers used in IR systems now typically have several GB of main memory, sometimes tens of GB. Available disk space is several (2 - 3) orders of magnitude larger.
- Fault tolerance is very expensive : It's much cheaper to use many regular machines rather than one fault tolerant machine.

Blocked sort - based indexing

- As for large collections that require the use of secondary storage, the central requirement of an external sorting algorithm is that it minimizes the number of random disks seeks during sorting, sequential disk reads are far faster than seeks. The Blocked Sort - Based Indexing algorithm (BSBI) is a solution.
- In BSBI
  i) Segments the collection into parts of equal size,
  ii) Sorts the termID-docID pairs of each part in memory,
  iii) Stores intermediate sorted results on disk, and
  iv) Merges all intermediate results into the final index.
- The time complexity of BSBI is O(T logT), because the step with the highest time complexity is sorting, and [T] is an upper bound for the number of items we must sort (i.e., the number of termID-docID pairs).

Scanned by TapScanner

- But the actual indexing time is usually dominated by the time it takes to parse the documents and to do the final merge.
- Each item in the list, which records that a term appeared in a document is conventionally called a posting. The list is then called a postings list or inverted list, and all the postings lists taken together are referred to as the postings.

Single - pass in - memory indexing

- Blocked sort - based indexing has excellent scaling properties, but it needs a data structure for mapping terms to termIDs. For very large collections, this data structure does not fit into memory. A more scalable alternative is Single - Pass In - Memory Indexing (SPIMI).
- SPIMI uses terms instead of termIDs, writes each block's dictionary to disk, and then starts a new dictionary for the next block. SPIMI can index collections of any size as long as there is enough disk space available.
- There are two advantages : It is faster because there is no sorting required, and it saves memory because we keep track of the term a postings list belongs to, so the termIDs of postings need not be stored.
- The time complexity of SPIMI is O[T] because no sorting of tokens is required, and all operations are at most linear in the size of the collection.

Distributed indexing

- Collections are often so large that we cannot perform index construction efficiently on a single machine. Web search engines, therefore, use distributed indexing algorithms for index construction.
- The distributed index construction method is an application of MapReduce, a general architecture for distributed computing. MapReduce is designed for large computer clusters.
- In the map phase, the collections were split into many parts. Parsing, BSBI, or SPIMI can be applied to each part. In reduce phase, the segments were combined together.
- The time to create a distributed index based on MapReduce architecture for Reuters-RCV1 may include map process time and reduce process time.
- In terms of the map phase, the operation can be broken down into reading, sorting, and writing. And for the reduce phase, it also needs to read, sort, and write.

Total time = Read time + Sort time + Write time

Scanned by TapScanner

2.4.1 Dynamic Indexing

- Up to now, we have assumed that collections are static. This is suitable for collections that change infrequently or never.
- But most collections are modified frequently with documents being added, deleted, and updated. This means that new terms need to be added to the dictionary, and postings lists need to be updated for existing terms.
- The simplest way to achieve this is to periodically reconstruct the index from scratch. This is a good solution if the number of changes over time is small and a delay in making new documents searchable is acceptable and if enough resources are available to construct a new index while the old one is still available for querying.
- This means that the dictionary and postings lists have to be modified :
  a) Postings updates for terms already in dictionary
  b) New terms added to dictionary
- If we store each postings list as a separate file, then the merge simply consists of extending each postings list of the main index by the corresponding postings list of the auxiliary index.
- In this scheme, the reason for keeping the auxiliary index is to reduce the number of disk seeks required over time. Updating each document separately requires up to Mbase disk seeks, where Mbase is the average size of the vocabulary of documents in the collection.
- The one-file-per-postings-list scheme is infeasible because most file systems cannot efficiently handle very large numbers of files.

2.4.2 Positional Indexes

- Here, for each term in the index vocabulary, we store postings of the form docID : <position1, position2, .> Each posting will also usually record the term frequency.
- To process a phrase query, you still need to access the inverted index entries for each distinct term.
- Example :
  to, 993433 :
  { 1, 6 : _7, 18, 33, 72, 86, 231};
  2, 5 : {1, 17, 74, 222, 255};
  4, 5 : {8, 16, 190, 429, 433};
  5, 2 : {363, 367};
  7, 3 : {13, 23, 191}; . . .}

Scanned by TapScanner

be, 178239 :

{ 1, 2 : {17, 25};

4, 5 : {17, 191, 291, 430, 434};

5, 3 : {14, 19, 101}; . . . }

- We start with the least frequent term and then work to further restrict the list of possible candidates. In the merge operation, the same general technique is used as before, but rather than simply checking that both terms are in a document, you also need to check that their positions of appearance in the document are compatible with the phrase query being evaluated. This requires working out offsets between the words.

2.5 Multiple Choice Questions

Q.1 _____ is a list of words that we should ignore when processing documents, since they give no useful information about content.

a Verb
b Startlist
c Stop words
d All of these

Q.2 The automatic conflation operation is also called ----------.

a stemming
b recall
c precision
d all of these

Q.3 Zipf's law is a law about the frequency distribution of _____ in a language.

a character
b words
c sentences
d all of these

Q.4 Tokenization breaks the raw text into words, sentences called _____.

a index
b stemming
c tokens
d verb

Q.5 Zipf's law states that :

a The frequency of a word type is proportional to its rank by frequency.
b The frequency of a word type is inversely proportional to its rank by frequency.
c The frequency of a word token is proportional to its rank by frequency.
d The frequency of a word token is inversely proportional to its rank by frequency

Scanned by TapScanner

Q.6 **Stemming algorithms are basic _____ algorithms, which depend on rules which identify affixes that can be stripped.**

a word handling

b paragraph handling

c string handling

d all of these

Answer Keys for Multiple Choice Questions

Q.1	c	Q.2	a	Q.3	b	Q.4	c
Q.5	b	Q.6	c				

❑❑❑

Scanned by TapScanner

3 Retrieval Models

Syllabus

Boolean, vector space, TFIDF, Okapi, probabilistic, language modeling, latent semantic indexing. Vector space scoring. The cosine measure. Efficiency considerations. Document length normalization. Relevance feedback and query expansion. Rocchio.

Contents

3.1 *Information Retrieval Models*

3.2 *Term Weighting*

3.3 *Inverted File*

3.4 *Language Model based IR*

3.5 *Probabilistic IR*

3.6 *Latent Semantic Indexing*

3.7 *Relevance Feedback and Query Expansion*

3.8 *Query Generation Probability*

3.9 *Document Length Normalization*

3.10 *Multiple Choice Questions*

Scanned by TapScanner

3.1 Information Retrieval Models

- An information retrieval system is a software program that stores and man-ages information on documents, often textual documents but possibly multimedia.
- Reason behind use of models in information retrieval is that models guide research and provide the means for academic discussion. Models can also serve as a blueprint to implement an actual retrieval system.
- A model of information retrieval predicts and explains what a user will find relevant given the user query. The correctness of the model's predictions can be tested in a controlled experiment.

3.1.1 Basic Concept

- Each document represented by a set of representative keywords or index terms. An index term is a document word useful for remembering the document main themes.
- Usually, index terms are nouns because nouns have meaning by themselves. However, search engines assume that all words are index terms.
- Not all terms are equally useful for representing the document contents: less frequent terms allow identifying a narrower set of documents. The importance of the index terms is represented by weights associated to them
- Model is an idealization or abstraction of an actual process. Mathematical models are used to study the properties of the process, draw conclusions, and make predictions. Conclusions derived from a model depend on whether the model is a good approximation of the actual situation. Statistical models represent repetitive processes; make predictions about frequencies of interesting events.

3.1.2 Boolean Model

- The Boolean model is the first model of information retrieval and probably also the most criticized model. It is based on set theory and Boolean algebra.
- It is based on a binary decision criterion without any notion of a grading scale. Boolean expressions have precise semantics. It is not simple to translate an information need into a Boolean expression. It can be represented as a disjunction of conjunction vectors(in disjunctive normal form-DNF).

D : Set of words (indexing terms) present in a document. Each term is either present (1) or absent (0).

Q : A Boolean expression. The terms are index terms and operators are AND, OR and NOT.

Scanned by TapScanner

F : Boolean algebra over sets of terms and sets of documents.

R : A document is predicted as relevant to a query expression if and only if it satisfies the query expression

$$((text \vee information) \wedge retrieval \wedge \neg theory)$$

- Each query term specifies a set of documents containing the term :
  a. AND ($\wedge$) : The intersection of two sets b. OR ($\vee$) : The union of two sets
  c. NOT ($\neg$) : Set inverse, or really set difference.

Boolean Relevance example :

$$((text \vee information) \wedge retrieval \wedge \neg theory)$$

It gives following list :

"Information Retrieval"

"Information Theory"

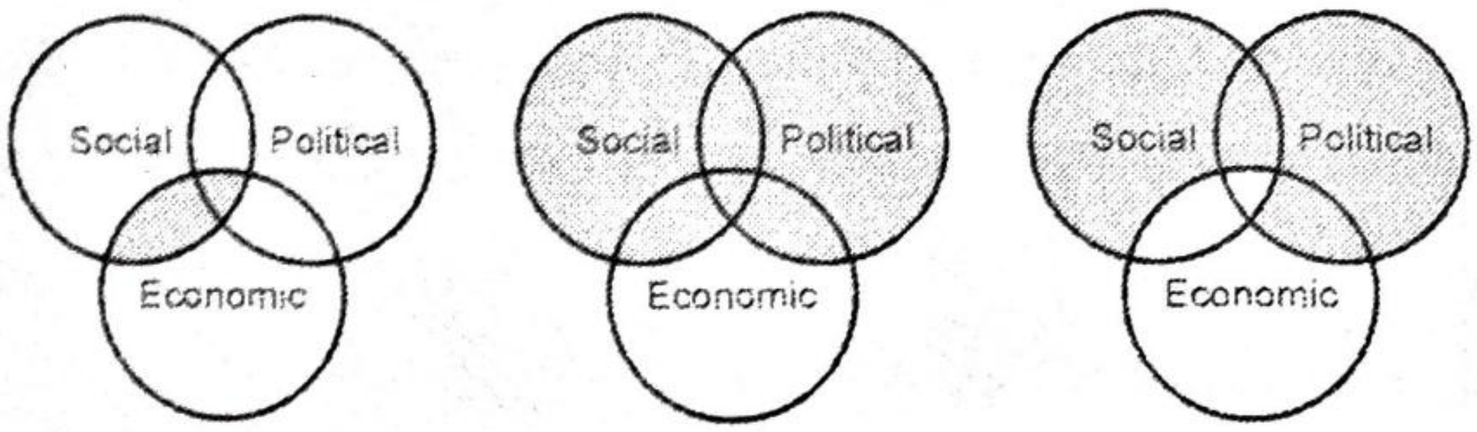
"Modern Information Retrieval: Theory and Practice"

"Text Compression"

Implementing the Boolean Model :

- First, consider purely conjunctive queries ($t_a \wedge t_b \wedge t_c$).
- It only satisfied by a document containing all three terms.
- If $D(t_a) = \{d \mid t_a \in d\}$, then the *maximum possible* size of the retrieved set is the size of the smallest $D(t_a)$.
- $|D(t_a)|$ is the length of the inverted list for t_a.
- For instance, the query **social AND economic** will produce the set of documents that are indexed both with the term social and the term economic, i.e. the intersection of both sets.
- Combining terms with the OR operator will define a document set that is bigger than or equal to the document sets of any of the single terms.
- So, the query social OR political will produce the set of documents that are indexed with either the term social or the term political or both, i.e. the union of both sets.
- This is visualized in the Venn diagrams of Fig. 3.1.1 in which each set of documents is visualized by a disc.
- The intersections of these discs and their complements divide the document collection into 8 non-overlapping regions, the unions of which give 256 different Boolean combinations of "social, political and economic documents". In Fig. 3.1.1 the retrieved sets are visualized by the shaded areas.

Scanned by TapScanner



Social AND Economic | Social OR Political | (Social OR Political) AND NOT (Social AND Economic)

Fig. 3.1.1 Boolean combinations of sets visualized as Venn diagrams

Advantages :

a. Clean formalism

b. Simplicity

c. It is very precise in nature. The user exactly gets what is specified.

d. Boolean model is still widely used in small scale searches like searching emails, files from local hard drives or in a mid-sized library.

Disadvantages :

a. It is not simple to translate an information need into a Boolean expression

b. Exact matching may lead to retrieval of too few or too many documents.

c. The retrieved documents are not ranked.

d. The model does not use term weights.

3.1.3 Vector Model

- Assign non-binary weights to index terms in queries and in documents. Compute the similarity between documents and query. The index terms in the query are also weighted

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

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

- Term weights are used to compute the degree of similarity between documents and the user query. Then, retrieved documents are sorted in decreasing order.
- They considered the index representations and the query as vectors embedded in a high dimensional Euclidean space, where each term is assigned a separate dimension. The similarity measure is usually the cosine of the angle that separates the two vectors $\vec{d}$ and $\vec{q}$.
- The cosine of an angle is 0 if the vectors are orthogonal in the multidimensional space and 1 if the angle is 0 degrees. The cosine formula is given by :

Scanned by TapScanner

$$\text{score}(\vec{d}, \vec{q}) = \frac{\sum_{k=1}^{m} d_k \cdot q_k}{\sqrt{\sum_{k=1}^{m} (d_k)^2} \cdot \sqrt{\sum_{k=1}^{m} (q_k)^2}}$$

- The metaphor of angles between vectors in a multidimensional space makes it easy to explain the implications of the model to non-experts. Up to three dimensions, one can easily visualise the document and query vectors. Fig. 3.1.2 shows a query and document representation in the vector space model.

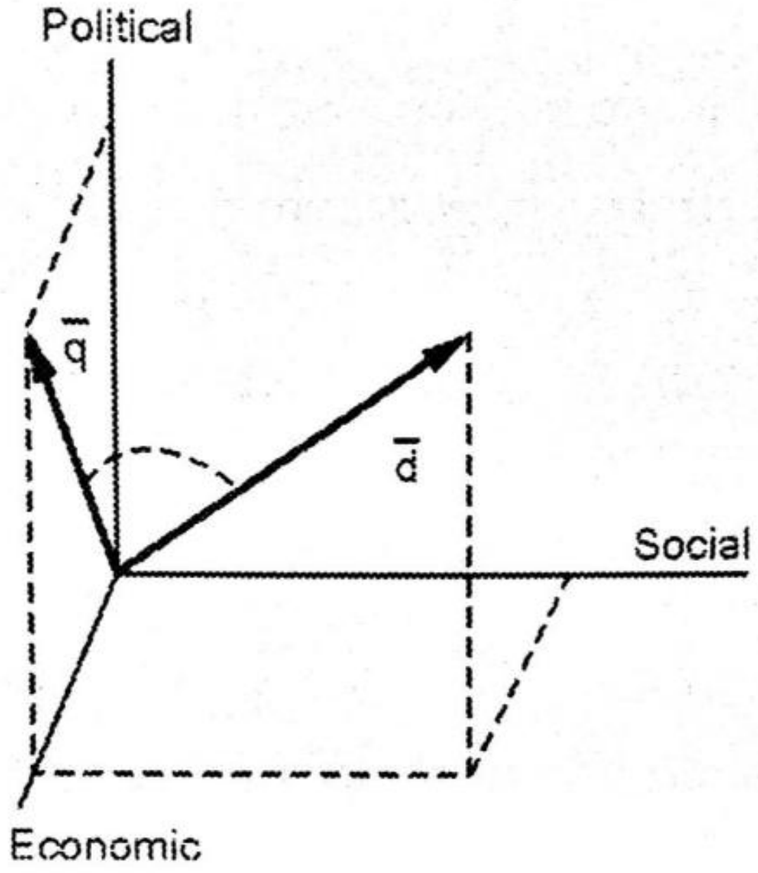


Fig. 3.1.2 Query and document representation in the vector space model

- Measuring the cosine of the angle between vectors is equivalent with normalizing the vectors to unit length and taking the vector inner product. If index representations and queries are properly normalised, then the vector product measure of equation 1 does have a strong theoretical motivation. The formula then becomes :

$$\text{score}(\vec{d}, \vec{q}) = \sum_{k=1}^{m} n(d_k) \cdot n(q_k)$$

$$\text{where } n(v_k) = \frac{v_k}{\sqrt{\sum_{k=1}^{m} (v_k)^2}}$$

- We think of the documents as a collection C of objects and think of the user query as a specification of a set A of objects. In this scenario, the IR problem can be reduced to the problem of determine which documents are in the set A and which ones are not (i.e. the IR problem can be viewed as a clustering problem).

1. **Intra-cluster** : One needs to determine what are the features which better describe the objects in the set A.
2. **Inter-cluster** : One needs to determine what are the features which better distinguish the objects in the set A.

Scanned by TapScanner

tf : Inter-clustering similarity is quantified by measuring the raw frequency of a term k_i inside a document d_j , such term frequency is usually referred to as the tf factor and 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. This frequency is often referred to as the **inverse document frequency.**

Advantages :

1. Its term-weighting scheme improves retrieval performance.
2. Its partial matching strategy allows retrieval of documents that approximate the query conditions.
3. Its cosine ranking formula sorts the documents according to their degree of similarity to the query.

Disadvantages :

1. The assumption of mutual independence between index terms.
2. It cannot denote the "clear logic view" like Boolean model.

3.2 Term Weighting

- A computerized information retrieval system can actually operate to retrieve some information, that information must have already been stored inside the computer. Originally it will usually have been in the form of documents.
- The computer, however, is not likely to have stored the complete text of each document in the natural language in which it was written. It will have, instead, a document representative which may have been produced from the documents either manually or automatically.
- Term weighting is an important aspect of modern text retrieval systems. Terms are words, phrases, or any other indexing units used to identify the contents of a text. Since different terms have different importance in a text, an important indicator - the term weight - is associated with every term.
- The retrieval performance of the information retrieval systems is largely dependent on similarity measures. Furthermore, a term weighting scheme plays an important role for the similarity measure.
- There are three components in a weighting scheme :

$$a_{ij} = gi * t_{ij} * d_j$$

Scanned by TapScanner

Where g_i is the global weight of the ith term,

t_{ij} is the local weight of the ith term in the jth document,

d_j is the normalization factor for the jth document.

3.2.1 TF-IDF Weighting

- Term Frequency (TF) : Frequency of occurrence of query keyword in document
- Inverse Document Frequency (IDF) : How many documents the query keyword occurs in.
- Inverse Document Frequency (IDF) is a popular measure of a word's importance. It's defined as the logarithm of the ratio of number of documents in a collection to the number of documents containing the given word. This means rare words have high IDF and common words have low IDF.
- Term frequency is a measure of the importance of terms *i* in document *j*.
- Inverse document frequency is a measure of the *general* importance of the term.
- High term frequency for "apple" means that apple is an important word in a specific document. But high document frequency (low inverse document frequency) for "apple", given a particular set of documents, means that apple is not all that important overall, since it is in all of the documents.
- The weight increases as the number of documents in which the term appears decreases. High value indicates that the word occurs more often in this document than average.
- The term frequency $tf_{t,d}$ of term t in document d is defined as the number of times that t occurs in d.
- A document with tf = 10 occurrences of the term is more relevant than a document with tf = 1 occurrence of the term. But not 10 times more relevant. Relevance does not increase proportionally with term frequency.
- The document frequency is the number of documents in the collection that the term occurs in. We define the idf weight of term t as follows :

$$\text{idf weight } (idf_t) = \log 10 \frac{N}{df_t}$$

here N is the number of documents in the collection

- The tf-idf weight of a term is the product of its tf weight and its idf weight

$$W_{t,d} = (1 + \log tf_{t,d}) \cdot \log \frac{N}{df_t}$$

Stop lists and Stemming :

- **Stoplists :** This is a list of words that we should ignore when processing documents, since they give no useful information about content.

Scanned by TapScanner

- **Stemming :** This is the process of treating a set of words like "fights, fighting, fighter, ..." as all instances of the same term - in this case the stem is "fight".

3.2.2 Luhn's Ideas

- Luhn's determined significant index words by their frequency counts in the document text, as representation or characterization of a document. The result, a derived list of index terms, can be called keywords or terms for each document.
- Luhn's idea was that words with high and low frequency are too common and too rate to contribute significantly to the content of a document, and that only words with medium frequency are significant.
- The idea of using frequency to measure word significance is based on the fact that a writer normally repeats certain words when writing on a subject.
- The more a specific word is found in a document, the more significance may be assigned to these words. He also takes common words in consideration. Such words must be present to tie the words together, but the type of significance does not reside in such words.
- These common words must be segregated and excluded from being considered as significant words by some method. As frequency has been proposed as a criterion, this step can be seen as an upper boundary of significant words.
- A lower boundary is needed too. This is to bracket the most useful range of words. Fig. 3.2.1 shows Luhn's idea.

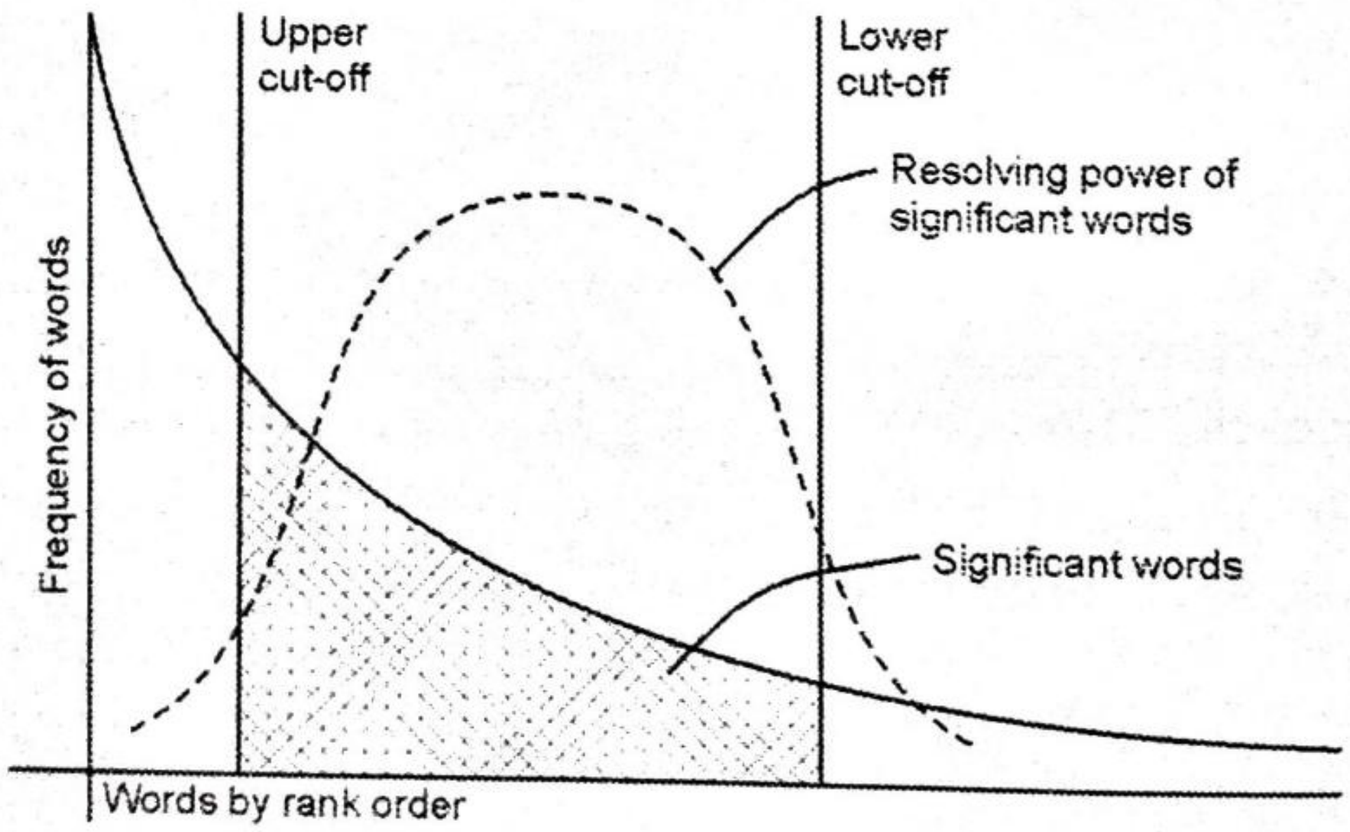


Fig. 3.2.1 Luhn's idea

- The frequency curve actually demonstrates Zipf's law. With this law Zipf stated that the product of the frequency of occurrences of words and the rank order is approximately constant.
- Luhn's method for each document :
  1. Filter terms in the document using a list of stopwords.
  2. Normalize terms by stemming like differentiate, different, differently.

Scanned by TapScanner

3. Calculate frequencies of normalized terms.
4. Remove non-frequent terms.

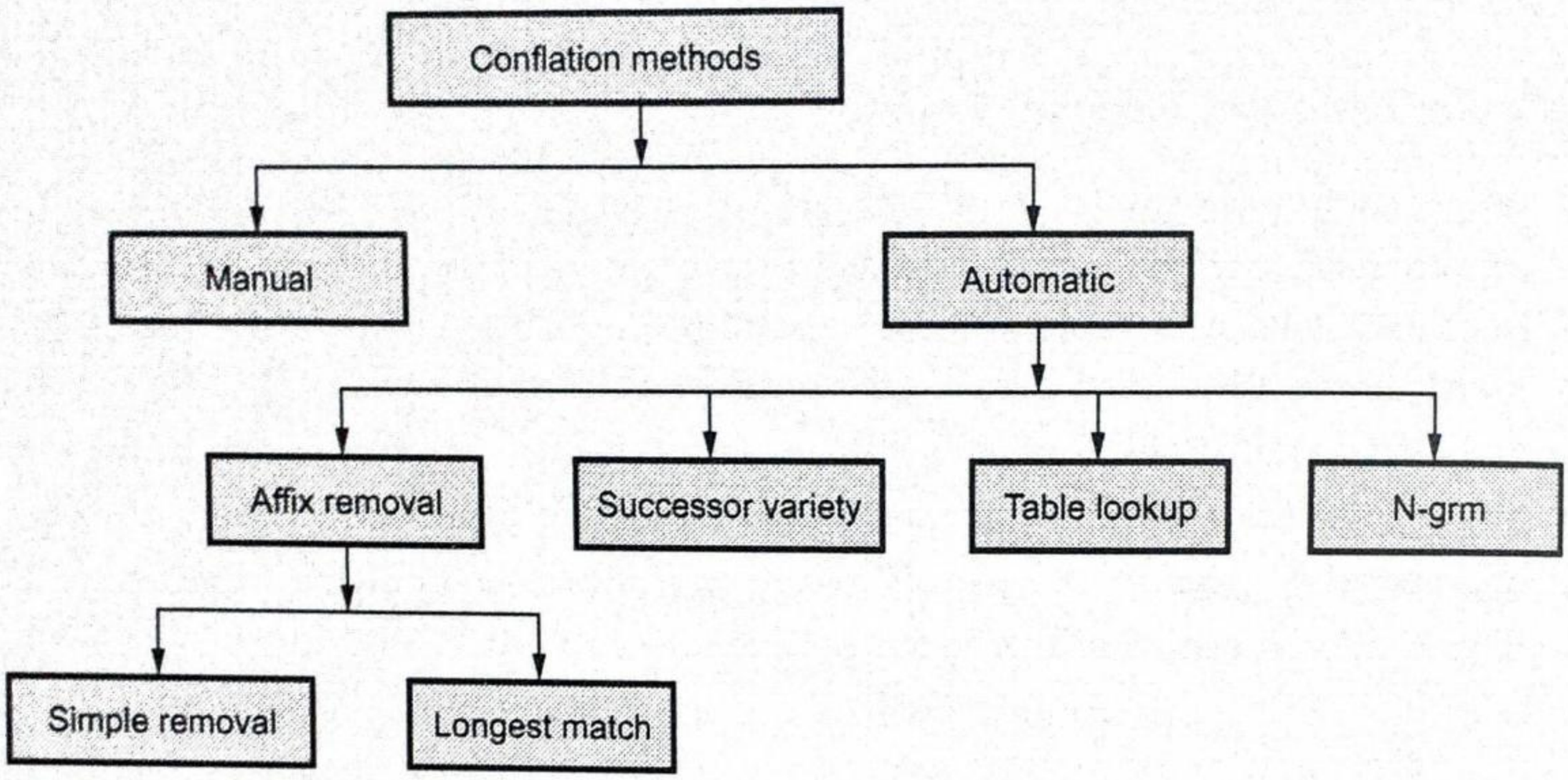
- Problem with Luhn's model is that the cut off points to be determined empirically, but trial and error.

3.2.3 Conflation Algorithm

- Information retrieval is essentially a matter of deciding which documents in a collection should be retrieved to satisfy a user's need for information. The user's information need is represented by a query or profile, and contains one or more search terms, plus some additional information such importance weights.
- The retrieval decision is made by comparing the terms of the query with the index terms (important words or phrases) appearing in the document itself.
- The decision may be binary (retrieve/reject), or it may involve estimating the degree of relevance that the document has to the query.
- For example morphological variants (e.g., COMPUTATIONAL, COMPUTER, COMPUTERS, COMPUTING etc.) are generally the most common, with other sources including valid alternative spellings, mis-spellings, and variants arising from transliteration and abbreviation.
- A number of stemming algorithms, or stemmers, have been developed, which attempt to reduce a word to its stem or root form. Stemming is the process for reducing inflected words to their stem, base or root form, generally a written word form. The process of stemming is often called conflation. These programs are commonly referred to as stemming algorithms or stemmers.
- Word conflation is the process by which a group of words that have the same meaning are reduced to a single term. For example : COLLECT, COLLECTED, COLLECTING, COLLECTION, COLLECTIONS , can all be conflated to the common term COLLECT.
- The key terms of a query or document are represented by stems rather than by the original words. This not only means that different variants of a term can be conflated to a single representative form. It also reduces the dictionary size. The number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time.
- Conflation algorithms are classified into two main classes : Stemming algorithms which are language dependent and string-similarity algorithms which are language independent.
- Stemming algorithms typically try to reduce related words to the same base (or stem).System will usually consist of three parts :

Scanned by TapScanner

1. Removal of high frequency words,
2. Suffix stripping,
3. Detecting equivalent stems.



- The removal of high frequency words, 'stop' words or 'fluff' words is one way of implementing Luhn's upper cut-off. This is normally done by comparing the input text with a 'stop list' of words which are to be removed.
- The advantages of the process are not only that non-significant words are removed and will therefore not interfere during retrieval, but also that the size of the total document file can be reduced by between 30 % to 50 %.
- The second stage, suffix stripping, is more complicated. A standard approach is to have a complete list of suffixes and to remove the longest possible one. For example, we may well want UAL removed from FACTUAL but not from EQUAL.
- Suffix stripping algorithms do not rely on a lookup table that consists of inflected forms and root form relations. Instead, a typically smaller list of "rules" is stored which provide a path for the algorithm, given an input word form, to find its root form. Some examples of the rules include :
  1. If the word ends in 'ed', remove the 'ed'
  2. If the word ends in 'ing', remove the 'ing'
  3. If the word ends in 'ly', remove the 'ly'
- Following are the simple steps of conflation algorithm :
  1. Open and read each input file and create a single index file.
  2. Remove or filer out all stop words.
  3. Remove all suffixes/affixes from each word if present.

Scanned by TapScanner

4. Count frequencies of occurrences for each root word from 3.
5. Apply porters rules/algorithm for each root word from 3 and store in index file.

Affix removal stemmers :

a) If a word ends in "ies" but not "eies" or "aies" then replace "ies" with "y"

b) If a word ends in "es" but not "aes", "ees", or "oes" then replace "es" with "e"

c) If a word ends in "s", but not "us" or "ss" then replace "s" with NULL.

3.2.4 Cosine Similarity

- This metric is frequently used when trying to determine similarity between two documents. Since there are more words that are in common between two documents, it is useless to use the other methods of calculating similarities.
- As a result, the likelihood that two documents do not share the majority is very high and does not create a satisfactory metric for determining similarities.
- In this similarity metric, the attributes (or words, in the case of the documents) is used as a vector to find the normalized dot product of the two documents. By determining the cosine similarity, the user is effectively trying to find cosine of the angle between the two objects.
- For cosine similarities resulting in a value of 0, the documents do not share any attributes (or words) because the angle between the objects is 90 degrees.
- Cosine similarity is expressed as a mathematical equation :

$$\text{similarity } (x, y) = \cos(\theta) = \frac{x \cdot y}{||x|| * ||y||}$$

- Cosine is a normalized dot product. Documents are ranked by decreasing cosine value.
- A term that appears in many documents should not be regarded as more important than one that appears in few documents.
- A document with many occurrences of a term should not be regarded as less important than a document with few occurrences of the term

3.3 Inverted File

- Each document is assigned a list of keywords or attributes. Each keyword (attribute) is associated with operational relevance weights.
- An inverted file is the sorted list of keywords (attributes), with each keyword having links to the documents containing that keyword.
- Penalty : The size of inverted files ranges from 10 % to 100 % of more of the size of the text itself. It need to update the index as the data set changes.

Scanned by TapScanner

- Inverted file is composed of two elements :
  a. Vocabulary b. Occurrences
- **Vocabulary** is the set of all different words in the text. For each such word a list of all the text positions where the word appears is stored. The set of all those lists is called the occurrences .
- A controlled vocabulary which is the collection of keywords that will be indexed. Words in the text that are not in the vocabulary will not be indexed. A list of stop-words that for reasons of volume will not be included in the index.
- A set of rules that decide the beginning of a word or a piece of text that is indexable. A list of character sequences to be indexed (or not indexed).
- Fig. 3.3.1 shows a sample text with inverted index file. Each entry in the vocabulary has **the word**, a **pointer into the postings structure** and **word metadata.**

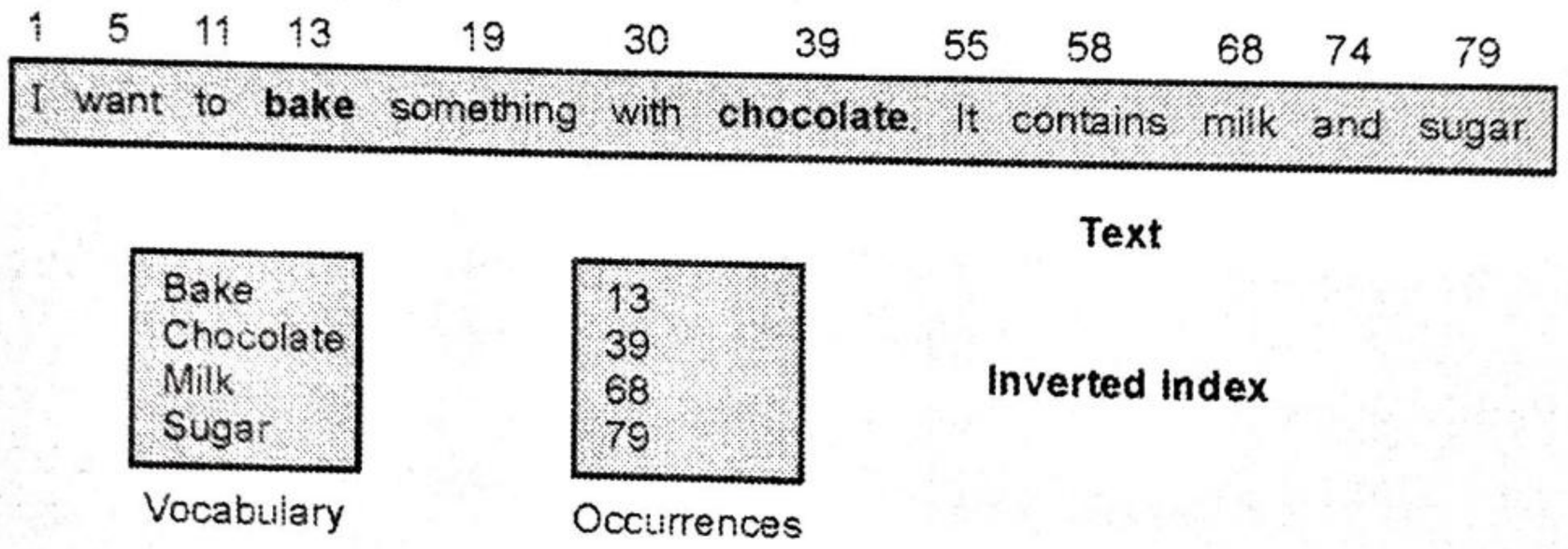


Fig. 3.3.1 Sample text with inverted index file

- Every query goes here first
  a. Keep in memory or a separate file b. Search should be fast
  c. Support prefix or pattern matching d. Support updating
- The positions can refer to words or characters. The word positions simplify phrase and proximity queries, while character positions facilities direct access to the matching text positions. Time needed to access posting lists is a function of their length and their allocation.
- The space required for the **vocabulary** is rather small. According to Heaps law the vocabulary grows as O(nB), where B is a constant between 0 and 1 dependent on the text. The **occurrences** required much more space. Each word appearing in the text is referenced once in that structure, the extra space is O(n).
- Block addressing is used to reduce the space requirement. The text is divided in blocks and the occurrences point to the blocks where the word appears. The classical indices which point to the exact occurrences are called **full inverted indices.**

Scanned by TapScanner

- The definition of inverted files does not require that the addresses in the directory are in any order. However, to facilitate operations such as conjunction ('and') and disjunction ('or') on any two inverted lists, the addresses are normally kept in record number order. This means that 'and' and 'or' operations can be performed with one pass through both lists. The penalty we pay is of course that the inverted file becomes slower to update.
- Fig. 3.3.2 shows the sample text split into blocks and an inverted index using block addressing built on it.

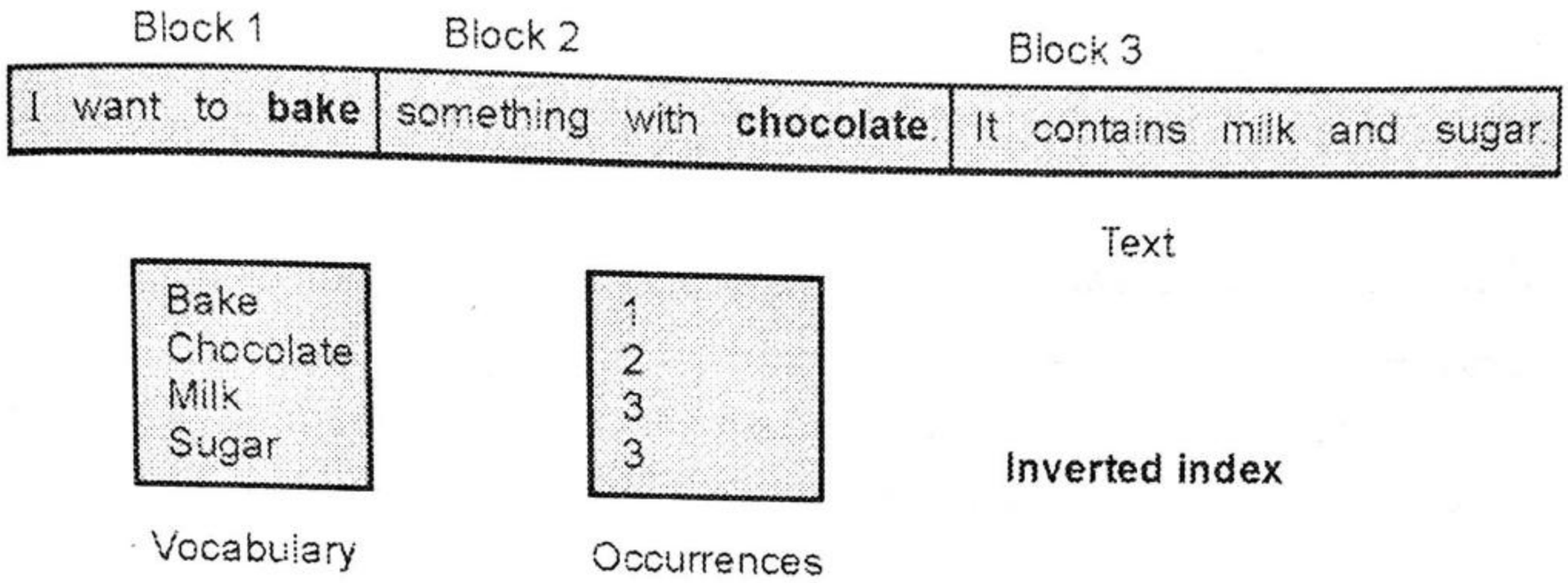


Fig. 3.3.2 Sample text split into blocks and an inverted index using block addressing

- The blocks can be of fixed size or they can be defined using the natural division of the text collection into files, documents, web pages etc. Concept of block improves the retrieval efficiency.

3.3.1 Searching

- The search algorithm on an inverted index follows three steps :
  1. **Vocabulary search :** The words and pattern presents in the query are isolated and searched in the vocabulary.
  2. **Retrieval of occurrences :** The list of the occurrences of all the words found is retrieved.
  3. **Manipulation of occurrences :** The occurrences are processed to solve phrases, proximity or Boolean operations. If block addressing is used it may be necessary to directly search the text to find the information missing from the occurrences.
- Structures used in inverted files are sorted arrays, hashing structures, Tries (digital search trees) and combinations of these structures. Single word queries can be searched using any suitable data structure to speed up the search.
- If the index stores character positions the phrase query cannot allow the separators to be disregarded and the proximity has to be defined in terms of character distance.

Scanned by TapScanner

3.3.2 Construction

- Building and maintaining an inverted index is a relatively low cost task. An inverted index on a text of "n" characters can be built in O(n) time. All the vocabulary is kept into the trie data structure, storing for each word a list of its occurrences.
- Each word of the text is read and searched in the trie. If it is found, it is added to the trie with an empty list of occurrences. Once it is in the trie, the new position is added to the end of its list of occurrences. Fig. 3.3.3 shows building an inverted index for the sample text.
- For simplicity, split the index into two files.
  1. First file contains list of occurrences that are stored contiguously. This file is typically called a **posting file.**
  2. Second file : The vocabulary is stored in lexicographical order and for each word; a pointer to its list in the first file is also included. This allows the vocabulary to be kept in memory at search time.

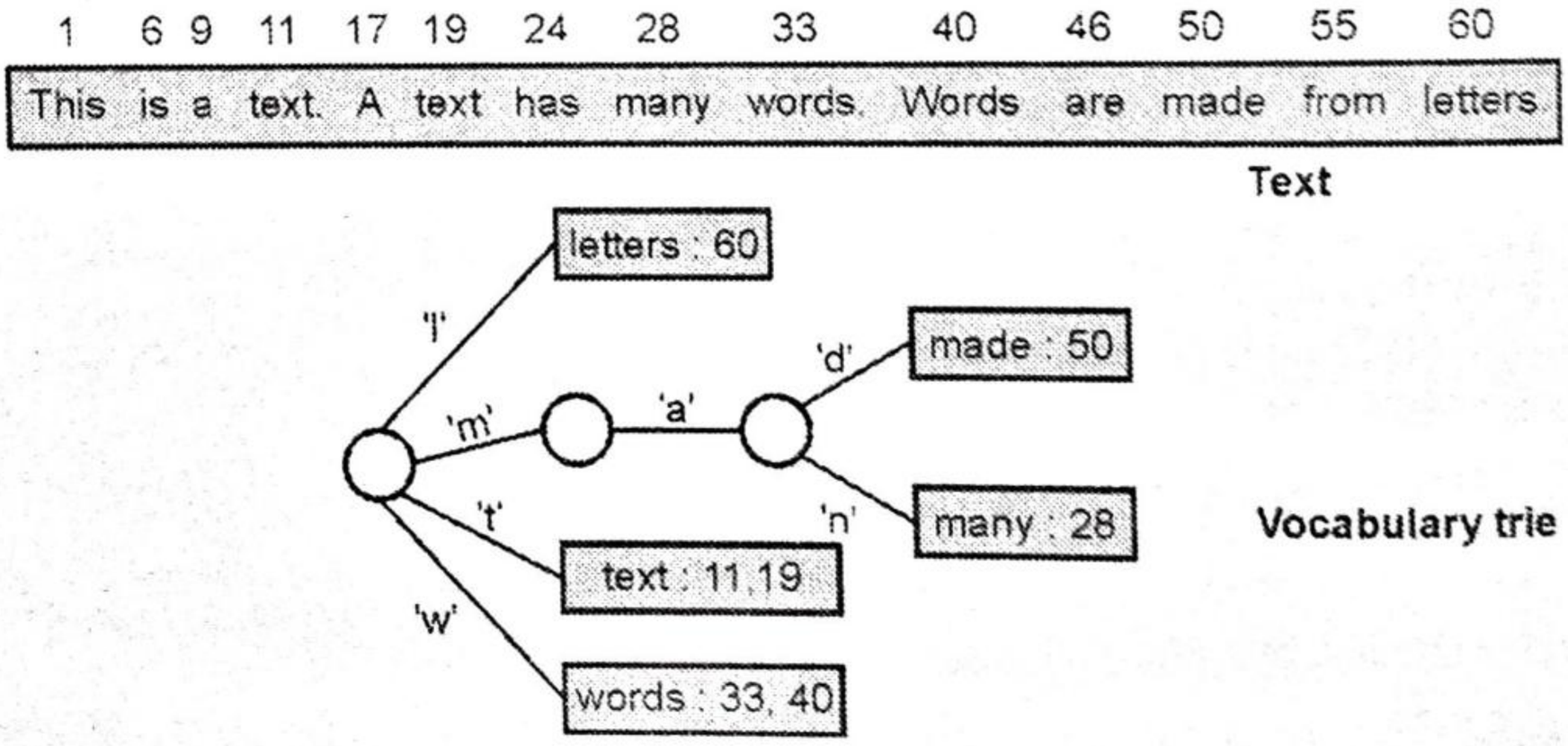


Fig. 3.3.3 Building an inverted index for the sample text

3.4 Language Model based IR

- A language model is a probabilistic mechanism for generating text. Language models estimate the probability distribution of various natural language phenomena.
- LMs assign probabilities to sequences of tokens. It is based on previous word histories.

Next Word Prediction :

a. Stocks ...

Scanned by TapScanner

b. Stocks plunged this ....

c. Stocks plunged this morning, despite a cut in interest rates

d. Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall ...

e. Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall Street began

f. Stocks plunged this morning, despite a cut in interest rates by the Reserve bank of India, as stock market began trading for the first time since last ...

- Idea of Language Model : A document is a good match to a query if the document model is likely to generate the query.
- Language modeling tries to capture the notion that some text is more likely than others. It does so by estimating the probability P (s) of any text s.

Finite automata and language models

- Informally, a state machine that comprehensively captures all possible states and transitions that a machine can take while responding to a stream(or machine can take while responding to a stream of input symbol. Using document language models to assign likelihood scores to queries.
- The notion of a language model is inherently probabilistic. A language model is a function that puts a probability measure over strings drawn from some vocabulary.

Types of Language Model

1. Unigram Language Model
2. N-gram Language Models

Unigram Language Model

- A unigram (1-gram) language model makes the strong independence assumption that words are generated independently from a multinomial distribution θ.
- For document classification, people found that unigram is often sufficient. Trigrams have been the standard for speech recognition.
- Example :

This is a sentence

Unigrams:

This,

is,

a,

sentence

Scanned by TapScanner

N-gram Language Models

- Unigram is a special case when n = 1. Common names include bigram (n = 2) and trigram (n = 3). N-grams traditionally do not run across sentences.

N = 2 (bigrams)	N = 3 (trigrams)
This is a sentence	This is a sentence
Bigrams : This is, is a, a sentence	**Trigrams :** This is a, is a sentence

- Language models are very useful in a broad range of applications, the most obvious perhaps being speech recognition and machine translation.

3.5 Probabilistic IR

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

Definition

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

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

Scanned by TapScanner

- Using Bayes' rule

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

where :

a. P(R) stands for the probability that a document randomly selected from the entire collection is relevant.

b. $P(d_j|R)$ stands for the probability of randomly selecting the document dj from the set R of relevant documents.

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

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

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

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

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

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

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

$\because \quad Pr(\vec{d}_j|R) + Pr(\vec{d}_j|\bar{R}) = 1$

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

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

Scanned by TapScanner

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

Disadvantages :

1. The need to guess the initial relevant and non-relevant sets.
2. Term frequency is not considered.
3. Independence assumption for index terms.

3.6 Latent Semantic Indexing

- Information is retrieved by literally matching terms in documents with those of a query. However, lexical matching methods can be inaccurate when they are used to match a user's query.
- Latent Semantic Indexing (LSI) tries to overcome the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. LSI assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice.
- LSI is an indexing and retrieval method that uses a mathematical technique called Singular Value Decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text.
- Latent Semantic Indexing is a technique that projects queries and documents into a space with "latent" semantic dimensions.
- It is statistical method for automatic indexing and retrieval that attempts to solve the major problems of the current technology. It is intended to uncover latent semantic structure in the data that is hidden. It creates a semantic space wherein terms and documents that are associated are placed near one another

General idea

1. Map documents (and terms) to a low-dimensional representation.
2. Design a mapping such that the low-dimensional space reflects semantic associations.
3. Compute document similarity based on the inner product in the latent semantic space

Goals

1. Similar terms map to similar location in low dimensional space
2. Noise reduction by dimension reduction

Scanned by TapScanner

- Latent semantic indexing is the application of a particular mathematical technique, called Singular Value Decomposition or SVD, to a word-by-document matrix. SVD is a least-squares method.

Example : following are the technical memo titles and construct document matrix

1. c1 : Human machine interface for ABC computer applications
2. c2 : A survey of user opinion of computer system response time
3. c3 : The EPS user interface management system
4. c4 : System and human system engineering testing of EPS
5. c5 : Relation of user perceived response time to error measurement
6. m1 : The generation of random, binary, ordered trees
7. m2 : The intersection graph of paths in trees
8. m3 : Graph minors IV: Widths of trees and well-quasi-ordering
9. m4 : Graph minors: A survey

Solution : First we construct the document matrix

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

(Note : 1 represents that word is present in that technical memo titles)

Scanned by TapScanner

3.7 Relevance Feedback and Query Expansion

- Users have no detailed knowledge of collection makeup and the retrieval environment. Most users often need to reformulate their queries to obtain the results of their interest.
- Thus, the first query formulation should be treated as an initial attempt to retrieve relevant information. Documents initially retrieved could be analyzed for relevance and used to improve the initial query.
- Fig. 3.7.1 shows relevance feedback on initial query.

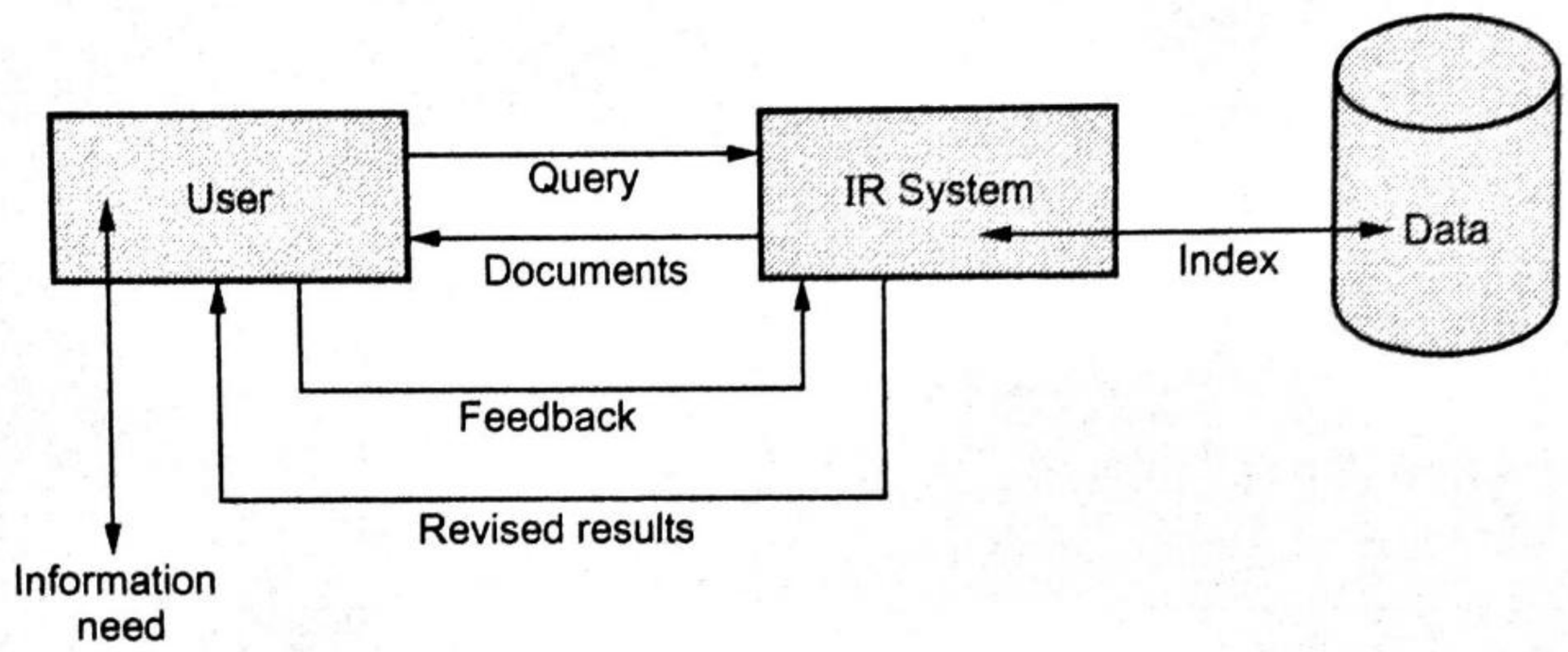


Fig. 3.7.1 (a) Relevance feedback

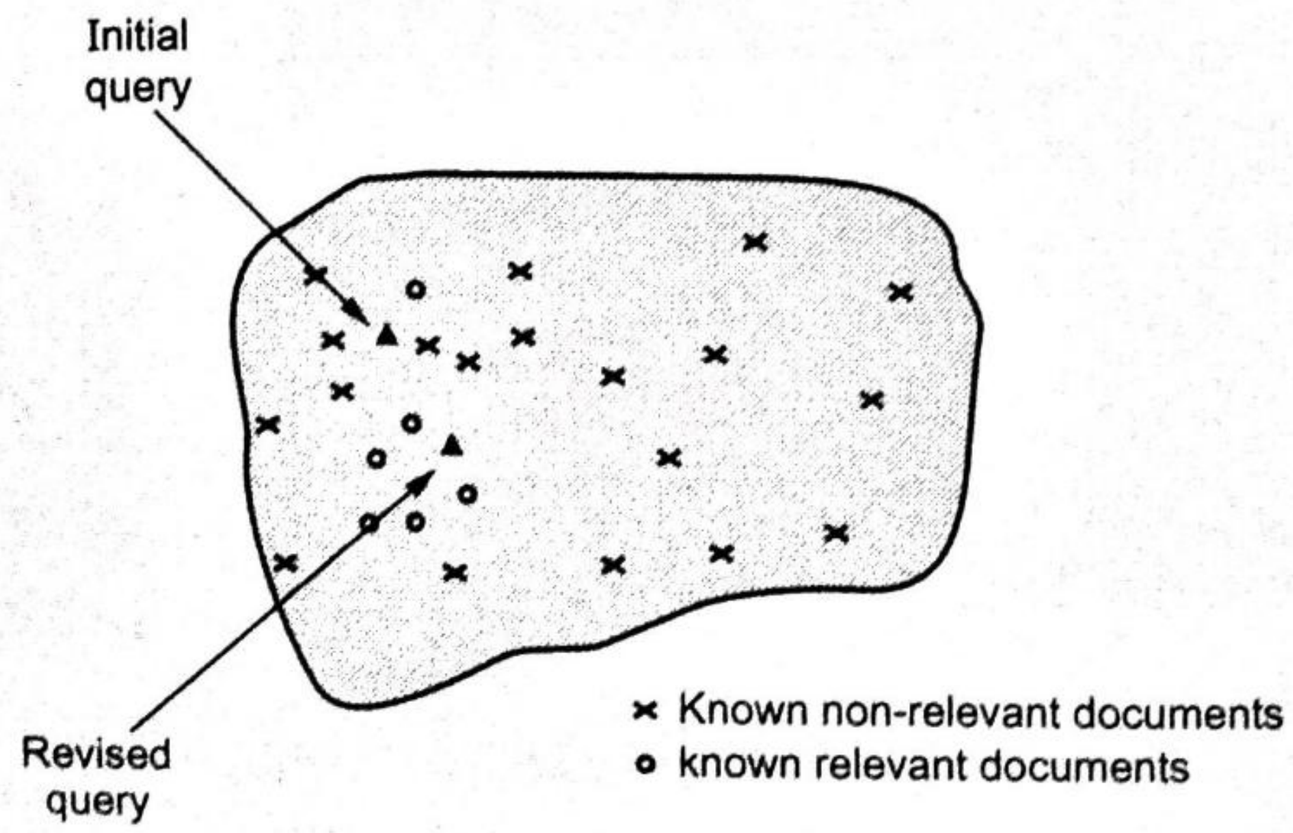


Fig. 3.7.1 (b) Relevance feedback on initial query

- The process of query modification is commonly referred as
  1. Relevance feedback, when the user provides information on relevant documents to a query.
  2. Query expansion, when information related to the query is used to expand it

Scanned by TapScanner

- Here we refer to both of them as feedback methods.
- Two basic approaches of feedback methods :
  1. **Explicit feedback** : The information for query reformulation is provided directly by the users. However, collecting feedback information is expensive and time consuming.
  2. **Implicit feedback** : The information for query reformulation is implicitly derived by the system.
- There are two basic approaches for compiling implicit feedback information :
  1. Local analysis, which derives the feedback information from the top ranked documents in the result set.
  2. Global analysis, which derives the feedback information from external sources such as a thesaurus.
- **Goal of relevance feedback :**
  1. Add query terms and adjust term weights.
  2. Improve ranks of known relevant documents.
  3. Other relevant docs will also be ranked higher.

Relevance feedback : Basic idea

- The user issues a short and simple query. The search engine returns a set of documents. User marks some docs as relevant, some as non-relevant.
- Search engine computes a new representation of the information need. Hope that it is better than the initial query. Search engine runs new query and returns new results. New results have better recall. Fig. 3.7.2 shows relevance and click feedback.

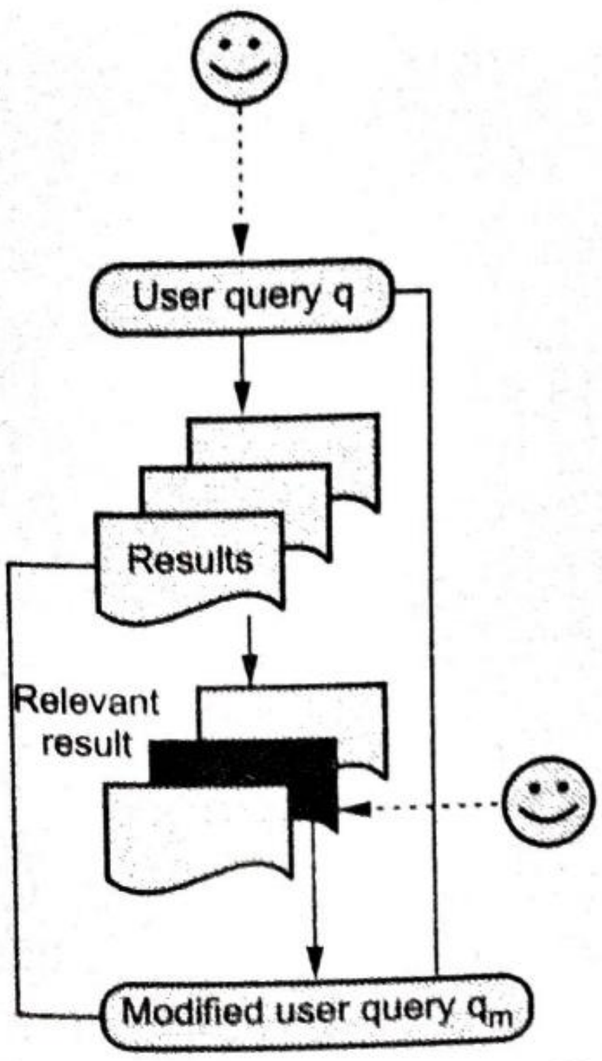


Fig. 3.7.2 (a) Relevance feedback

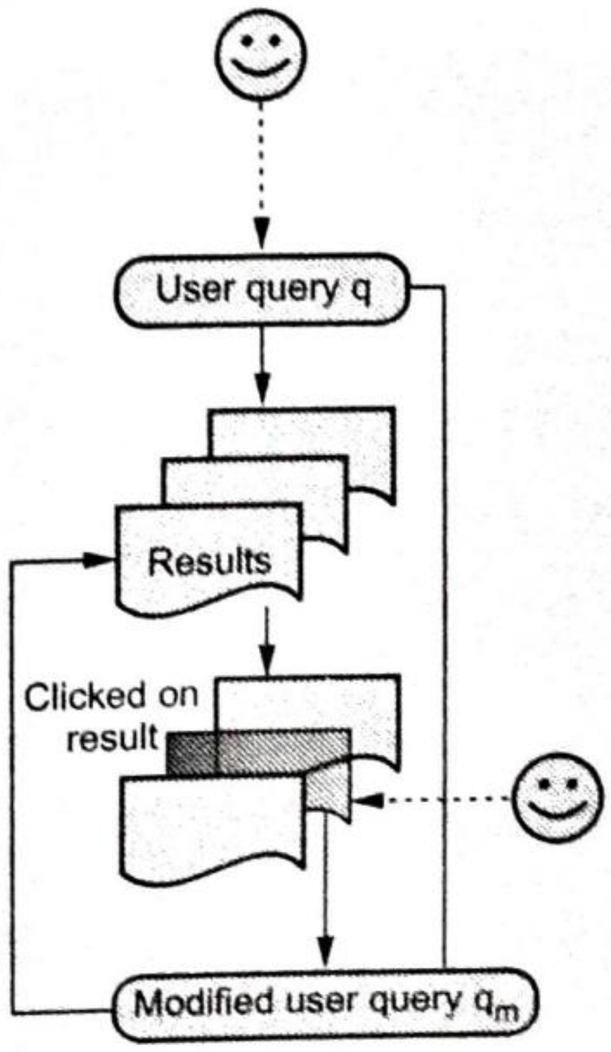


Fig. 3.7.2 (b) Click feedback

Scanned by TapScanner

- Characteristics of relevance feedback :
  1. It shields the user from the details of the query reformulation process.
  2. It breaks down the whole searching task into a sequence of small steps which are easier to grasp.
  3. Provide a controlled process designed to emphasize some terms (relevant ones) and de-emphasize others (non-relevant ones).

- Issues with relevance feedback

1. The user must have sufficient knowledge to form the initial query.
2. This does not work too well in cases like : Misspellings, CLIR and mismatch in user's and document's vocabulary (Burma vs. Myanmar).
3. Relevant documents have to be similar to each other while similarity between relevant and non-relevant document should be small.
4. Long queries generated may cause long response time.
5. Users are often reluctant to participate in explicit feedback.

- **Advantages of relevance feedback**

1. Relevance feedback usually improves average precision by increasing the number of good terms in the query.
2. It breaks down the search operation into a sequence of small search steps, designed to approach the wanted subject area gradually.
3. It provides a controlled query alteration process designed to emphasize some terms and to deemphasize others, as required in particular search environments.

- **Disadvantages of relevance feedback**

1. More computational work
2. Easy to decrease precision

3.7.1 Rocchio Method

- Rocchio's model is a classic framework for implementing relevance feedback via improving the query representation.
- Rocchio's algorithm is an iterative method which can generate successive queries based on an initial query and a set of relevant documents selected among the top k ranked documents.
- This algorithm uses vector space model to pick a relevance feedback theory.

Rocchio Properties :

1. Does not guarantee a consistent hypothesis.
2. Forms a simple generalization of the examples in each class (a prototype).

Scanned by TapScanner

3. Prototype vector does not need to be averaged or otherwise normalized for length since cosine similarity is insensitive to vector length.
4. Classification is based on similarity to class prototypes.

- Rocchio method is represented by following formula :

$$Q_1 = Q_0 + \beta \sum_{i=1}^{n_1} \frac{R_i}{n_1} - \gamma \sum_{i=1}^{n_2} \frac{S_i}{n_2}$$

where Q_0 = The vector for the initial query

R_i = The vector for the relevant document i

S_i = The vector for the non-relevant document i

n_1 = The number of relevant documents chosen

n_2 = The number of non-relevant documents chosen.

β and γ tune the importance of relevant and non relevant terms (in some studies best to set β to 0.75 and γ to 0.25).

- Rocchio's clustering algorithm was developed on the SMART project. It operates in three stages.
  1. In the first stage it selects a number of objects as cluster centers. The remaining objects are then assigned to the centers or to a 'rag-bag' cluster. On the basis of the initial assignment the cluster representatives are computed and all objects are once more assigned to the clusters. The assignment rules are explicitly defined in terms of thresholds on a matching function. The final clusters may overlap.
  2. The second stage is essentially an iterative step to allow the various input parameters to be adjusted so that the resulting classification meets the prior specification of such things as cluster size, etc. more nearly.
  3. The third stage is for 'tidying up'. Unassigned objects are forcibly assigned and overlap between clusters is reduced.

Example 3.7.1 *An IR system returns 8 relevant documents and 10 non-relevant documents. There are a total of 20 relevant documents in the collection. What is the precision of the system on this search and what is its recall ?*

Solution :

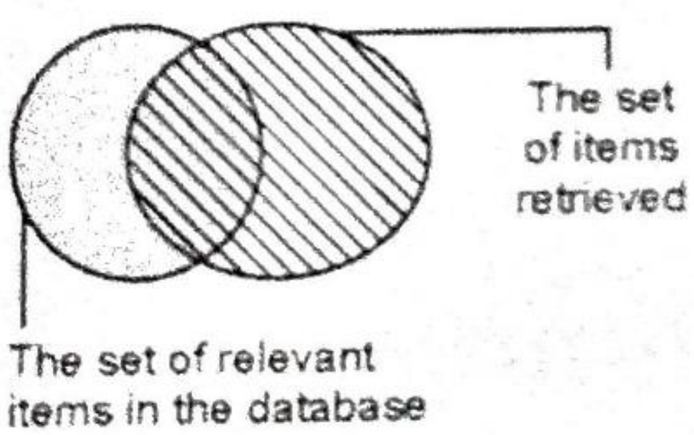
Precision = 8 / 18 = 0.44

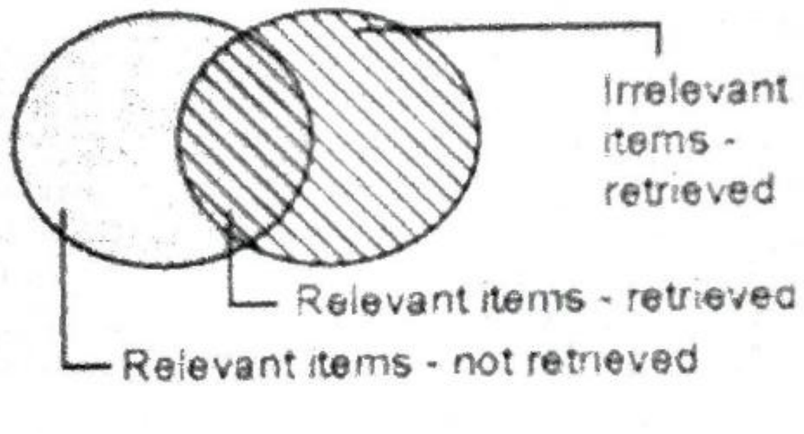
Recall = 8 / 20 = 0.40

Scanned by TapScanner

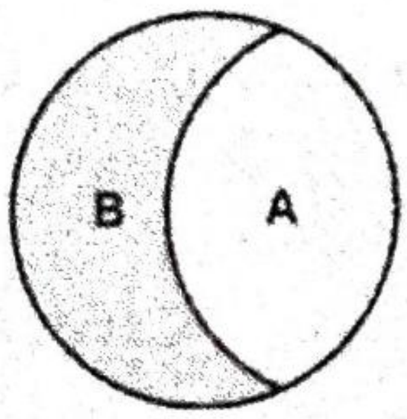
3.7.2 Precision and Recall

- **Relevance :** Relevance is a subjective notion. Different users may differ about the relevance or non-relevance of particular documents to given questions.
- In response to a query, an IR system searches its document collection and returns a ordered list of responses. It is called the retrieved set or ranked list. The system employs a search strategy or algorithm and measure the quality of a ranked list.
- A better search strategy yields a better ranked list and better ranked lists help the user fill their information need.
- Precision and recall are the basic measures used in evaluating search strategies. As shown in the first two figures, these measures assume :
  1. There is a set of records in the database which is relevant to the search topic.
  2. Records are assumed to be either relevant or irrelevant.
  3. The actual retrieval set may not perfectly match the set of relevant records.





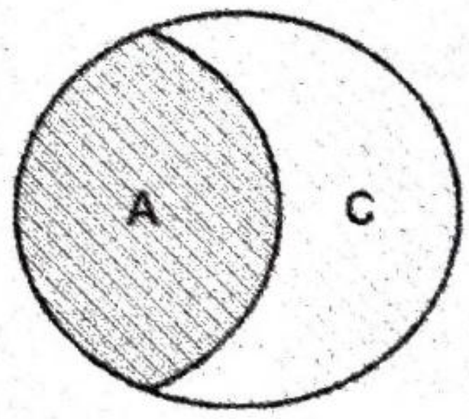
- **Recall** is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.



A = Number of relevant records retrieved.
B = Number of relevant records not retrieved.

$$\textbf{Recall} = \frac{A}{A+B} \times 100\ \%$$

Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.



A = Number of relevant records retrieved.
C = Number of irrelevant records retrieved.

Scanned by TapScanner

$$\text{Precison} = \frac{A}{A+C} \times 100\ \%$$

- As recall increases, the precision decreases and recall decreases the precision increases.

Example 3.7.2 *Assume the following :*

A database contains 80 records on a particular topic

A search was conducted on that topic and 60 records were retrieved.

Of the 60 records retrieved, 45 were relevant.

Calculate the precision and recall scores for the search.

Solution : Using the designations above :

A = The number of relevant records retrieved,

B = The number of relevant records not retrieved, and

C = The number of irrelevant records retrieved.

In this example A = 45, B = 35 (80 – 45) and C = 15 (60 – 45).

$$\text{Recall} = \frac{45}{45+35} \times 100\ \%$$

$$\text{Recall} = \frac{45}{80} \times 100\ \%$$

$$\textbf{Recall} = \textbf{56.25 \%}$$

$$\text{Precision} = \frac{A}{A+C} \times 100\ \%$$

$$\text{Precision} = \frac{45}{45+15} \times 100\ \% = \frac{45}{60} \times 100$$

$$\textbf{Precision} = \textbf{75 \%}$$

Example 3.7.3 *20 found documents, 18 relevant, 3 relevant documents are not found, 27 irrelevant are as well not found. Calculate the precision and recall and fallout scores for the search.*

Solution :

Precision : 18/20 = 90 %

Recall : 18/21 = 85.7 %

Fall-out : 2/29 = 6.9 %

- Recall is a non-decreasing function of the number of docs retrieved. In a good system, precision decreases as either the number of docs retrieved or recall increases. This is not a theorem, but a result with strong empirical confirmation.

Scanned by TapScanner

- The set of ordered pairs makes up the precision-recall graph. Geometrically when the points have been joined up in some way they make up the precision-recall curve. The performance of each request is usually given by a precision-recall curve. To measure the overall performance of a system, the set of curves, one for each request, is combined in some way to produce an average curve.
- Assume that set R_q containing the relevant document for q has been defined. Without loss of generality, assume further that the set R_q is composed of the following documents :

$$R_q = \{ d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123} \}$$

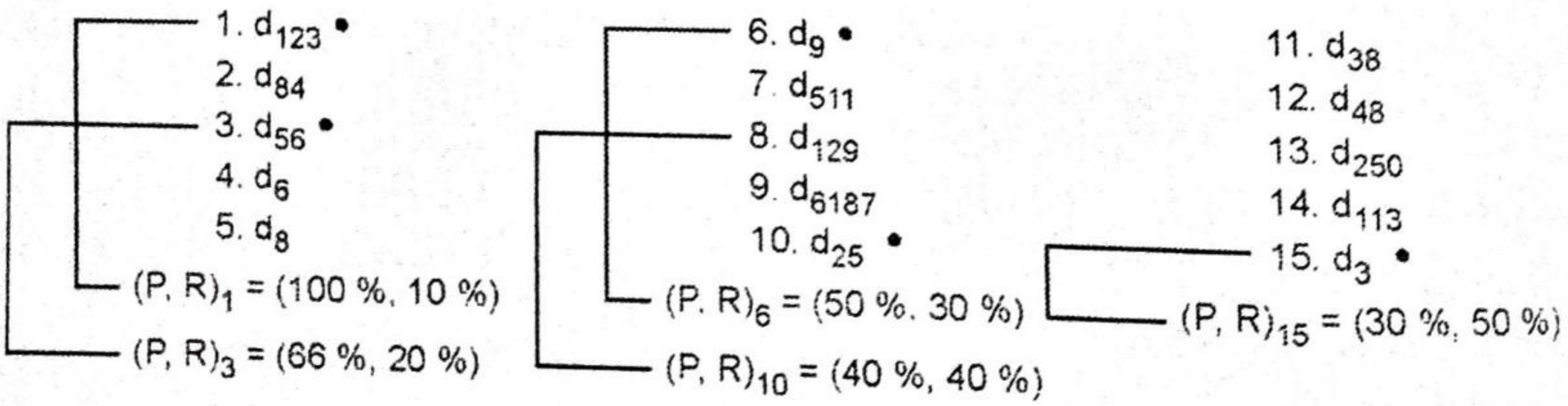
There are ten documents which are relevant to the query q.

- For the query q, a ranking of the documents in the answer set as follows.

Ranking for query q :

1. d_{123}	*	6. d_9	*	11. d_{38}	
2. d_{84}		7. d_{511}		12. d_{48}	
3. d_{56}	*	8. d_{129}		13. d_{250}	
4. d_6		9. d_{187}		14. d_{113}	
5. d_8		10. d_{25}	*	15. d_3	*

- The documents that are relevant to the query q are marked with star after the document number. Ten relevant documents, five included in Top 15.



- Fig 3.7.3 shows the curve of precision versus recall. By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a precision-recall curve.
- The precision versus recall curve is usually plotted based on 11 standard recall level : 0 %,10 %,....,100 %.
- In this example : The precisions for recall levels higher than 50 % drop to 0 because no relevant documents were retrieved. There was an interpolation for the

Scanned by TapScanner

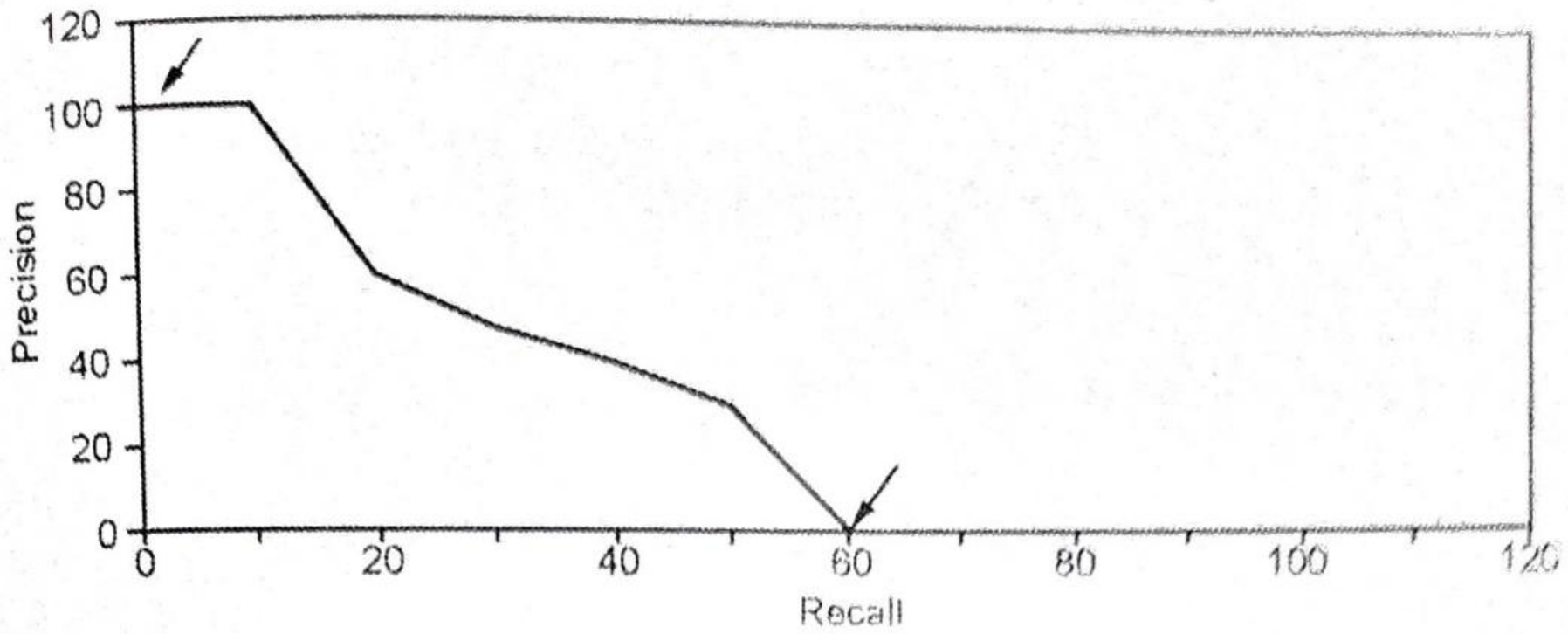
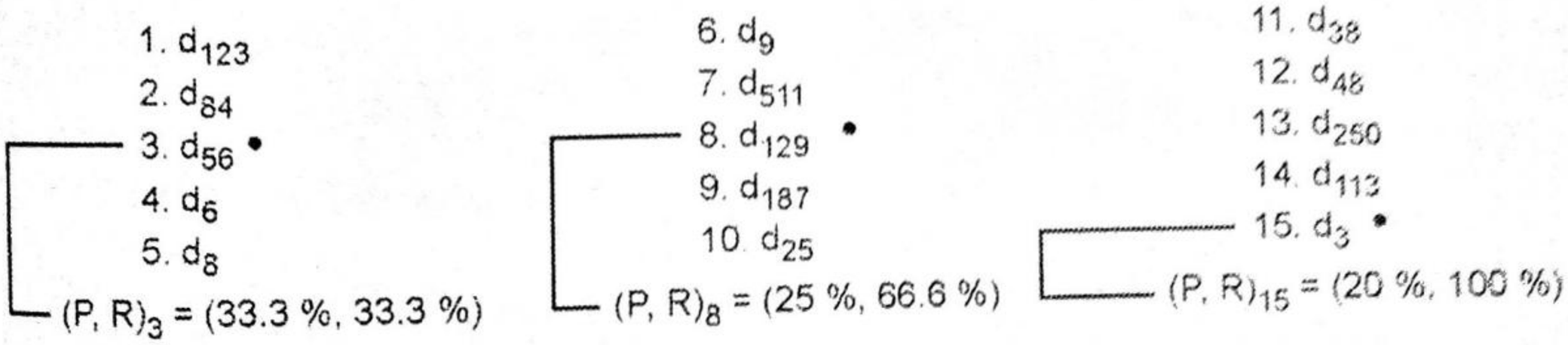


Fig. 3.7.3 Precision versus recall curve

- Since the recall levels for each query might be distinct from the 11 standard recall levels.

3.7.2.1 Interpolated Recall-Precision

- Idea : If locally precision increases with increasing recall, then you should get to count that. So you take the max of precisions to right of value.



- Consider again the set of 15 ranked documents. Assume that the set of relevant documents for the query q has changed and is now given by $R_q = \{d_3, d_{56}, d_{129}\}$
- The three relevant document :
- Interpolated precisions at standard recall levels :

$$\overline{P}(r_j) = \max_{r_j \le r \le r_{j+1}} P(r)$$

The jth standard recall level.

- Which means that the interpolated precision at the jth standard recall level is the maximum known precision at any recall level between the jth recall level and the $(j+1)^{th}$ recall level.
- At recall levels 0 %, 10 %, 20 % and 30 %, the interpolated precision is equal to 33.3 %.
- At recall levels 40 %, 50 %, 60 % the interpolated precision is equal to 25 %.

Scanned by TapScanner

- At recall levels 70 %, 80 %, 90 % and 100 %, the interpolated precision is equal to 20 %.
- Fig. 3.7.4 shows the curve for interpolated precision and recall.

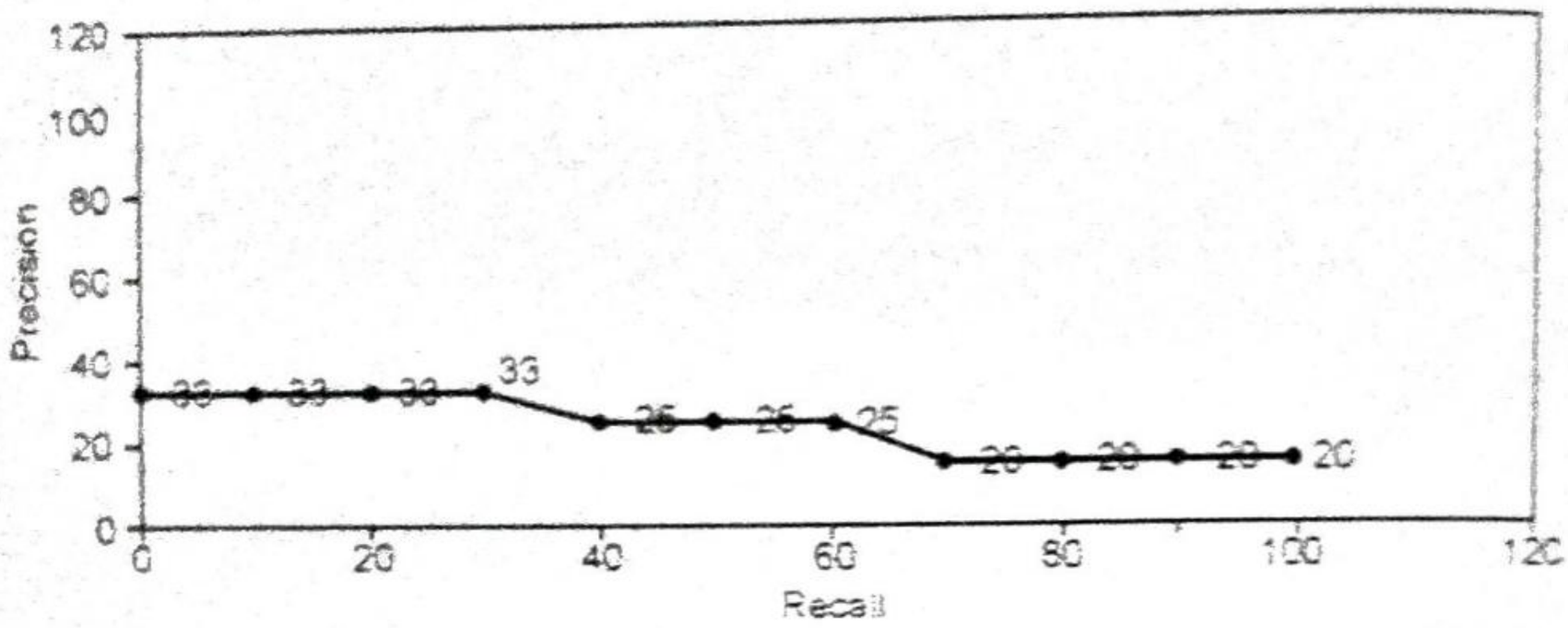


Fig. 3.7.4 Curve for interpolated precision and recall

- Following Fig. 3.7.5 shows the comparison between precision-recall curve and interpolated precision.

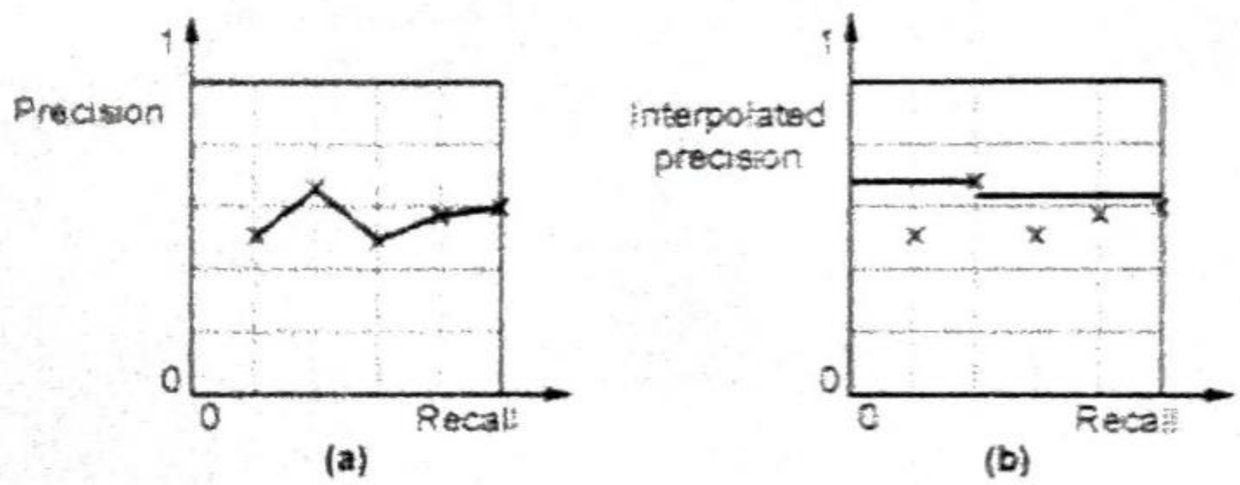


Fig. 3.7.5

Advantages of interpolated recall-precision

1. Simple, intuitive, and combined in single curve.
2. Provide quantitative evaluation of the answer set and comparison among retrieval algorithms.
3. A standard evaluation strategy for IR systems.

Disadvantages of interpolated recall-precision

1. Can not know true recall value except in small document collections.
2. Assume a strict document rank ordering.

- It is an experimental fact that average precision-recall graphs are monotonically decreasing. Consistent with this, a linear interpolation estimates the best possible performance between any two adjacent observed points. To avoid inflating the experimental results it is probably better to perform a more conservative interpolation.

Scanned by TapScanner

3.7.2.2 Mean Average Precision (MAP)

- Also called average precision at seen relevant documents. It determine precision at each point when a new relevant document gets retrieved.
- Average of the precision value obtained for the top k documents, each time a relevant doc is retrieved.
- Avoids interpolation, use of fixed recall levels. MAP for query collection is arithmetic averaging.
- Average precision-recall curves are normally used to compare the performance of distinct IR algorithms.
- Use P = 0 for each relevant document that was not retrieved. Determine average for each query, then average over queries :

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j} P(doc_i)$$

where

Q_j = Number of relevant document for query j.

N = Number of queries.

$P(doc_i)$ = Precision at i^{th} relevant document

Example 3.7.4

Query 1		
Rank	**Relev.**	**$P(doc_i)$**
1	X	1.00
2		
3	X	0.67
4		
5		
6	X	0.50
7		
8		
9		

Query 2		
Rank	**Relev.**	**P(doc)**
1	X	1.00
2		
3	X	0.67
4		
5		
6		
7		
8		
9		

Scanned by TapScanner

10	X	0.40
11		
12		
13		
14		
15		
16		
17		
18		
19		
20	X	0.50
AVG :		0.564

10		
11		
12		
13		
14		
15	X	0.2
AVG :		0.623

- MAP favors systems which return relevant documents fast.

Solution :

$$MAP = \frac{0.564 + 0.623}{2}$$

$$MAP = 0.594$$

- A necessary consequence of its monotonicity is that the average P-R curve will also be monotonically decreasing. It is possible to define the set of observed points in such a way that the interpolate function is not monotonically decreasing. In practice, even for this case, we have that the average precision-recall curve is monotonically decreasing.

Precision-recall appropriateness

- Precision and recall have been extensively used to evaluate the retrieval performance of IR algorithms. However, a more careful reflection reveals problems with these two measures :
- First, the proper estimation of maximum recall for a query requires detailed knowledge of all the documents in the collection.
- Second, in many situations the use of a single measure could be more appropriate.
- Third, recall and precision measure the effectiveness over a set of queries processed in batch mode.
- Fourth, for systems which require a weak ordering though, recall and precision might be inadequate.

Scanned by TapScanner

Single value summaries

- Average precision-recall curves constitute standard evaluation metrics for information retrieval systems. However, there are situations in which we would like to evaluate retrieval performance over individual queries. The reasons are twofold :
  1. First, averaging precision over many queries might disguise important anomalies in the retrieval algorithms under study.
  2. Second, we might be interested in investigating whether a algorithm outperforms the other for each query.
- In these situations, a single precision value can be used.

R-Precision

- If we have a known set of relevant documents of size Rel, then calculate precision of the top Rel docs returned.
- Let R be the total number of relevant docs for a given query. The idea here is to compute the precision at the Rth position in the ranking.
- For the query q1, the R value is 10 and there are 4 relevant among the top 10 documents in the ranking. Thus, the R-Precision value for this query is 0.4.
- The R-precision measure is a useful for observing the behavior of an algorithm for individual queries. Additionally, one can also compute an average R-precision figure over a set of queries.
- However, using a single number to evaluate a algorithm over several queries might be quite imprecise.

Precision histograms

- The R-precision computed for several queries can be used to compare two algorithms as follows :

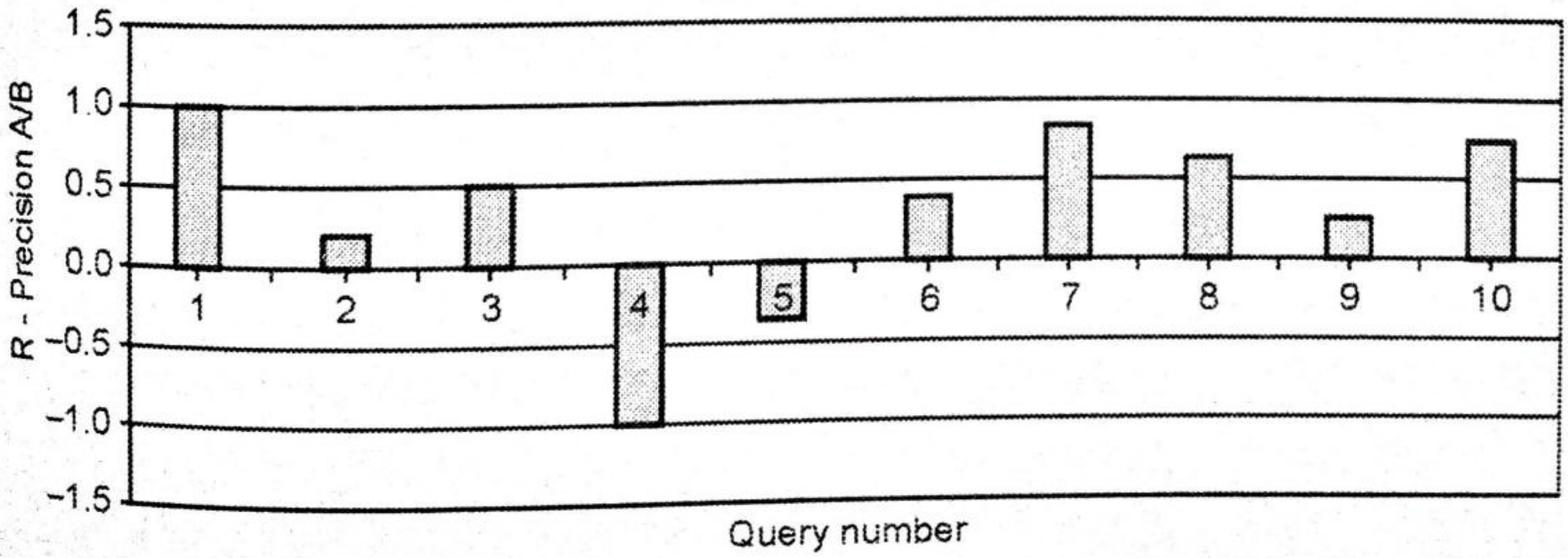


Fig. 3.7.6 Precision histograms

Scanned by TapScanner

- Let,

  $RP_A(i)$: R-precision for algorithm A for the i-th query

  $RP_B(i)$: R-precision for algorithm B for the i-th query

- Define, for instance, the difference :

  $RP_{A/B}(i) = RP_A(i) - RP_B(i)$

- A positive value of $RP_{A/B}(i)$ indicates a better retrieval performance by algorithm A while a negative value indicates a better retrieval performance by algorithm B. Fig. 3.7.6 shows the $RP_{A/B}(i)$ values for two retrieval algorithms over 10 example queries.
- The algorithm A performs better for 8 of the queries, while the algorithm B performs better for the other 2 queries.

3.7.3 Probability Relevance Feedback

- Guess a preliminary probabilistic description of *R* and use it to retrieve a first set of documents V. Interact with the user to refine the description: partition V into relevant and non-relevant VNR.
- If user has told us some relevant and some irrelevant documents, then we can proceed to build a probabilistic classifier, such as a Naive Bayes model :

  $$P(t_k|R) = |D_{rk}|/|D_r|$$

  $$P(t_k|NR) = |D_{nrk}|/|D_{nr}|$$

Where t_k is a term; D_r is the set of known relevant documents; D_{rk} is the subset that contain t_k; D_{nr} is the set of known irrelevant documents; D_{nrk} is the subset that contain t_k.

- Even though the set of known relevant documents is a perhaps small subset of the true set of relevant documents, if we assume that the set of relevant documents is a small subset of the set of all documents then the estimates given above will be reasonable. This gives a basis for another way of changing the query term weights.
- Reestimate p_i and r_i on the basis of these

  $$p_i^{(2)} = \frac{|V_i| + k p_i^{(1)}}{|V| + k}$$

- Repeat, thus generating a succession of approximations to R.

3.7.4 Pseudo Relevance Feedback

- Pseudo relevance feedback , also known as blind relevance feedback , provides a method for automatic local analysis. It automates the manual part of relevance feedback, so that the user gets improved retrieval performance without an extended interaction.

Scanned by TapScanner

- The method is to do normal retrieval to find an initial set of most relevant documents, to then assume that the top k ranked documents are relevant, and finally to do relevance feedback as before under this assumption.
- Relevance feedback is considered as pseudo (or blind) relevance feedback when there is an assumption that the top documents retrieved have a higher precision and that their terms represent the subject expected to be retrieved.
- In other words, it is assumed that the documents on the top of the retrieval list are relevant to the query, and information from these documents is extracted to generate a new retrieval set.
- Blind relevance feedback is typically performed in two stages :
  1. An initial search using the original topic statement is performed, after which a number of terms are selected from some number of the top-ranked documents. The selected terms are then weighted and then merged with the initial query to formulate a new query.
  2. Finally the reweighted and expanded query is submitted against the same collection to produce a final ranked list of documents. Obviously there are important choices to be made regarding the number of top-ranked documents to consider, and the number of terms to extract from those documents
- Following table shows this contingency table

	Relevant	**Not Relevant**	
In doc	R_t	$N_t - R_t$	N_t
Not in doc	$R - R_t$	$N - N_t - R + R_t$	$N - N_t$
	R	$N - R$	N

- The relevance weight is calculated using the assumption that the first 10 documents are relevant and all others are not. For each term in these documents the following weight is calculated :

$$w_t = \log \frac{\frac{R_t}{R - R_t}}{\frac{N_t - R_t}{N - N_t - R + R_t}}$$

Scanned by TapScanner

3.7.5 Indirect Relevance Feedback

- We can also use indirect sources of evidence rather than explicit feedback on relevance as the basis for relevance feedback. This is often called implicit (relevance) feedback.
- Implicit feedback is less reliable than explicit feedback, but is more useful than pseudo relevance feedback, which contains no evidence of user judgments.
- Moreover, while users are often reluctant to provide explicit feedback, it is easy to collect implicit feedback in large quantities for a high volume system, such as a web search engine.
- DirectHit ranked documents higher that users look at more often
- On the web, DirectHit introduced the idea of ranking more highly documents that users chose to look at more often. In other words, clicks on links were assumed to indicate that the page was likely relevant to the query.
- This approach makes various assumptions, such as that the document summaries displayed in results lists (on whose basis users choose which documents to click on) are indicative of the relevance of these documents.
- In the original DirectHit search engine, the data about the click rates on pages was gathered globally, rather than being user or query specific. This is one form of the general area of clickstream mining . Today, a closely related approach is used in ranking the advertisements that match a web search query

3.8 Query Generation Probability

- A query is sample of words drawn from a document based on the model defined for the document.
- Documents are then ranked based on their likelihood of giving (generating) that query. Document models that give a higher probability to the query indicate having more terms of the query (capturing the notion of TF)

$$\text{score}(Q, D) = p(Q|\theta\, D)$$

- Unigram query likelihood is given as

$$p(Q|\theta\, D) = \prod_{i=1}^{n} P(q_i|D)$$

$$P(q_i|D) = \frac{tf_{q_i,D}}{|D|}$$

- If a term in a query does not occur in a document, the whole similarity measure becomes zero.

Scanned by TapScanner

- To deal with the estimation problem and data sparsity, smooth the probability estimates by :

1. Lowering the probability estimate of the terms in document
2. Assigning probabilities to unseen terms in document

The probability of producing the query given the language model M_d of document d using Maximum Likelihood Estimation (MLE) and the unigram assumption is

$$\hat{P}(q|M_d) = \prod_{t \in q} \hat{P}_{mle}(t|M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}$$

where M_d is the language model of document d, $tf_{t,d}$ is the (raw) term frequency of term t in document d, and L_d is the number of tokens in document d.

- We just count up how often each word occurred, and divide through by the total number of words in the document d.
- In particular, some words will not have appeared in the document at all, but are possible words for the information need, which the user may have used in the query.
- If we estimate $\hat{P}(t|M_d) = 0$ for a term missing from a document d, then we get a strict conjunctive semantics : documents will only give a query non-zero probability if all of the query terms appear in the document.
- Zero probabilities are clearly a problem in other uses of language models, such as when predicting the next word in a speech recognition application, because many words will be sparsely represented in the training data.
- A non-occurring term is possible, but no more likely than would be expected by chance in the collection.
- If $tf_{t,d} = 0$,

$$P(t|M_d) = \frac{cf_t}{cs}$$

cf_t : Raw count of term t in collection

cs : Raw collection size (total number of tokens in the collection)

- Zero probabilities spell disaster so we need to smooth probabilities. There's a wide space of approaches to smoothing probability distributions to deal with this problem, such as adding 1, 1/2 or to ε counts, discounting, and interpolation.
- A simple idea that works well in practice is to use a mixture between the document multinomial and the collection multinomial distribution.

Scanned by TapScanner

3.9 Document Length Normalization

- Document length normalization adjusts the term frequency or the relevance score in order to normalize the effect of document length on the document ranking.
- Document length normalization of term weights is used to remove the advantages that the long documents have in retrieval over the short documents. Two main reasons that necessitate the use of normalization in term weights are :

1. Higher term frequencies : long documents usually use the same term repeatedly. As a result, the term frequency factors may be large for long documents, increasing the average contribution of its terms towards the query document similarity.
2. More terms : long documents also have numerous different terms.

3.10 Multiple Choice Questions

Q.1 List the classic Information retrieval model.

a Boolean b vector
c probabilistic d All of these

Q.2 In the vector model, document's and queries are represented as vectors in a t-dimensional space.

a Boolean b vector
c probabilistic d All of these

Q.3 The automatic conflation operation is also called _____.

a Stemming b recall
c precision d All of these

Q.4 _____ is the ratio of the number of relevant documents retrieved to the total number of documents retrieved.

a Recall b precision
c stemming d None

Q.5 Recall is the ratio of the number of _____ retrieved to the total number of relevant documents in the collection.

a Relevant document b Irrelevant document
c Structured document d Unstructured document

Scanned by TapScanner

Q.6 **_____ model is introduced by Roberston and Sparck Jones in 1976. It is also called Binary Independence Retrieval model.**

a Latent Semantic Indexing
b Boolean
c Vector
d Probabilistic

Q.7 **LSI is an indexing and retrieval method that uses a mathematical technique called _____ to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text.**

a Singular Value decomposition
b Multiple value decomposition
c Singular Value document
d Multiple value document

Q.8 **Rocchio's model is a classic framework for implementing _____ via improving the query representation.**

a Precision and recall
b Relevance Feedback
c Query formation
d None

Q.9 **Pseudo relevance feedback, also known as _____ relevance feedback, provides a method for automatic local analysis.**

a Probability
b Blind
c Mean Average Precision
d All of these

Q.10 **The representation of a set of documents as vectors in a common vector space is known as the _____ model.**

a Boolean
b document selection
c probabilistic
d vector

Q.11 **_____ ranking methods use the query to rank all documents in the order of relevance.**

a Text
b Word
c document
d information

Answer Keys for Multiple Choice Questions

Q.1	d	Q.2	b	Q.3	a	Q.4	b
Q.5	a	Q.6	d	Q.7	a	Q.8	b
Q.9	b	Q.10	d	Q.11	c		

❑❑❑

Scanned by TapScanner

4 Performance Evaluation

Syllabus

Evaluating search engines. User happiness, precision, recall, F-measure. Creating test collections: kappa measure, interjudge agreement.

Contents

4.1 Evaluating Search Engines

4.2 User Happiness

4.3 Creating Test Collections

4.4 Multiple Choice Questions

Scanned by TapScanner

4.1 Evaluating Search Engines

- The goal of an information retrieval system is to locate relevant documents in response to a user's query. Documents are typically retrieved as a ranked list, where the ranking is based on estimation of relevance.
- It is observed that a number of companies have developed various search engines such as Google, Lycos, Hotbot, Yahoo, Excite, AltaVista, etc.
- Six main measurable quantities of IRS :
  1) Coverage of the collection.
  2) Time lag, i.e., the average interval between the time the search request is made and the time an answer is given.
  3) Form of presentation of the output.
  4) Effort involved on the part of the user in obtaining answers to his search requests.
  5) Recall of the system.
  6) Precision of the system.
- There are generally millions of the sites pertaining in some manner to the queries submitted by the user, search engine will retrieve them, rank them on the basis of their relevance and present a huge list of documents.
- But studies show that approximately 80 % of web searchers never view more than first 10 results in a result's list. A search engine typically displays results page by page, perhaps 10 hits at a time.
- Many users tend to look at only the first page of results they get from a search engine, rather than scrolling through pages and pages of hits.
- A search engine is a document retrieval system designed to help find information stored in a computer system, such as on the WWW. The search engine allows one to ask for content meeting specific criteria and retrieves a list of items that match those criteria.
- Regardless of the underlying architecture, users specify keywords that match words in huge search engine databases, producing a ranked list of URLs and snippets of web-pages in which the keywords matched.
- Search engines are thus programmed in such a way that the first page shows a diversity of the most relevant links related to the keyword.
- To analyse the performance of the search engines, we used a metric, Ranked Precision (RP), which returns a number between 0 and 1. The RP measures the effectiveness of a search engine in terms of :

Scanned by TapScanner

a) First n documents, retrieved and ranked according to their relevance by search engine.

b) Number of relevant documents retrieved and their rank in first n documents.

- The formula to compute Ranked Precision (RP) metric is given below : j = n

$$RP = \frac{\sum_{j=1}^{j=n}((n+1)-r_j)) \times w_j}{(n(n+1)/2)}$$

Where n = First n documents displayed under consideration.

K = Number of relevant documents in n documents.

R_j = Rank of jth document.

W_j = Relevance score of jth document.

- Evaluation is key to building effective and efficient search engines.
- Effectiveness : Measure's ability to find right information. Compare ranking to user relevance feedback.
- Efficiency : Measures ability to do this quickly. Measure time and space requirements.
- Effectiveness, efficiency and cost are related. If we want a particular level of effectiveness and efficiency, this will determine the cost of the system configuration. Efficiency and cost targets may impact effectiveness.

4.1.1 TREC Collection

- TREC is a workshop series that provides the infrastructure for large-scale testing of (text) retrieval technology.
- The Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology (NIST) and U. S. Department of Defense, was started in 1992 as part of the TIPSTER Text program.
- Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies.
- In particular, the TREC workshop series has the following goals :
  1. To encourage research in information retrieval based on large test collections;
  2. To increase communication among industry, academia and government by creating an open forum for the exchange of research ideas;

Scanned by TapScanner

3. To speed the transfer of technology from research labs into commercial products by demonstrating substantial improvements in retrieval methodologies on real-world problems; and
4. To increase the availability of appropriate evaluation techniques for use by industry and academia, including development of new evaluation techniques more applicable to current systems.

- First TREC conference was held at NIST in November 1992, while the second TREC conference occurred in August 1993. Starting in TREC-3, a variety of other "tracks," were introduced.
- The tracks invigorate TREC by focusing research on new areas or particular aspects of text retrieval. The tracks represent the majority of the work done in TREC.
- Set of tracks in a particular TREC depends on :
  1. Interest of participants
  2. Appropriateness of task to TREC
  3. Needs of sponsors
  4. Resource constraints

Data collection

- The TREC collection has been growing steadily over the year. At TREC-3, the collection size was roughly 2 gigabytes while at TREC-6 it had gone up to roughly 5.8 gigabytes.
- The TREC collection is distributed in six CD-ROM disks of roughly 1 gigabyte of compressed text each. The documents come from the following sources :

WSJ	Wall Street Journal
AP	Associated Press
ZIFF	Computer Selects, Ziff-Davis
FR	Federal Register
DOE	US DOE Publications
SJMN	San Jose Mercury News
PAT	US Patents
FT	Financial TImes
CR	Congressional Record
FBIS	Foregin Broadcast Information service
LAT	LA Times

Scanned by TapScanner

Example information request

- The TREC collection includes a set of example information requests which can be used for testing a new ranking algorithm. Each request is a description of an information need in natural language. In the TREC nomenclature, each test information request is referred to as a topic.
- Following Fig. 4.1.1 shows the sample topic of TREC.

<top>

<num> Number: 351

<title> Falkland petroleum exploration

<desc> Description:

What information is available on petroleum exploration in the South Atlantic near the Falkland Islands?

<narr> Narrative:

Any document discussing petroleum exploration in the South Atlantic near the Falkland Islands is considered relevant. Documents discussing petroleum exploration in continental South America are not relevant.

</top>

Fig. 4.1.1 Sample topic of TREC

- The task of converting an information request into a system query must be done by the system itself and is considered to be an integral part of the evaluation performance.
- The topic numbered 1 to 150 was prepared for use with the TREC-1 and TREC-2 conference. The topics numbered 151 to 200 were prepared for use with the TREC-3 conference. It includes only three fields : Title, Description and Narrative.
- The topic numbered 201 to 250 was prepared for use with the TREC-4 conference. TREC-5 includes topic 251 to 300 and TREC-6 conference includes topic number 301 to 350.

Relevant documents for each example information request

- At the TREC conferences, the set of relevant documents for each example information request is obtained from a pool of possible relevant documents. This pool is created by taking the top K documents (usually, K = 100) in the ranking generated by the various participating retrieval systems.

Scanned by TapScanner

- The documents in the pool are then shown to human assessors who ultimately decide on the relevance of each document. This technique for accessing relevance is called the pooling method and based on two assumptions.
  1. The vast majority of the relevant documents is collected in the assembled pool.
  2. The documents which are not in the pool can be considered to be not relevant.

Benchmark tasks at the TREC conferences

- The TREC conferences include two main information retrieval tasks.

1. Ad hoc task :

- A set of new requests are run against a fixed document database. Example for this type is library. In library, where user is asking new queries against a set of static documents.
- The participant systems receive the test information requests and execute on a pre-specified document collection.

2. Routing task :

- A set of fixed requests are run against a database whose documents are continually changing. This is like filtering task in which the same questions are always being asked.
- The participant systems receive the test information requests and two distinct document collections.
- The first collection is used for training and allows the tuning of the retrieval algorithm. The second collection is used for testing the tuned retrieval algorithm.

TREC-6 secondary task

- Secondary task are as follows :
  1. Chinese : Ad hoc task in which both the documents and the topics are in Chinese.
  2. Filtering : Routing task in which the retrieval algorithm has only to decide whether a new incoming document is relevant or not. No ranking of the documents taken needs to be provided. The test data is processed in time stamp order.
  3. Interactive : Task in which a human searcher interacts with the retrieval system to determine the relevant documents. Documents are ruled relevant or not relevant.
  4. NLP : Task aimed at verifying whether retrieval algorithms based on natural language processing offer advantages when compared to the more traditional retrieval algorithms based on the index terms.

Scanned by TapScanner

5. Cross languages : Ad hoc task in which the documents are in one language but the topics are in a different language.
6. High precision : Task in which the user of a retrieval system is asked to retrieve ten documents that answer a given information request within five minutes.
7. Spoken document retrieval : Task in which the documents are written transcripts of radio broadcast news shows.
8. Very large corpus : Ad hoc task in which the retrieval systems have to deal with collections of size 20 gigabytes.

- In TREC-7, the NLP and Chinese secondary tasks were discontinued. TREC-7 also included a new task called Query Task in which several distinct query versions were created for each example information request.

Evaluation measures at the TREC conferences

TREC conferences uses four types of evaluations :

1. Summary table statistics
2. Recall-precision average
3. Document level averages
4. Average precision histograms

Summary table statistics

- Single value measures can also be stored in a table to provide a statistical summary regarding the set of all the queries in a retrieval task.
- For instance, these summary table statistics could include :
  1. The number of queries.
  2. Total number of documents retrieved by all queries.
  3. Total number of relevant documents which were effectively retrieved when all queries are considered.
  4. Total number of relevant documents which could have been retrieved by all queries.

Recall-precision average

- It consists of a table or graph with average precision at 11 standard recall levels.
- Since the recall levels of the individual queries are seldom equal to the standard recall levels, interpolation is used to define the precision at the standard recall levels.

Scanned by TapScanner

Document level averages

- Average precision is computed at specified document cutoff values.
- The average precision might be computed when 5, 10, 20, 100 relevant documents have been seen.
- The average R-precision value might also be provided.

Average precision histogram

- It consists of a graph which includes a single measure for each separate topic.
- It is difference between the R-precision for a target retrieval algorithm and the average R-precision computed from the results of all participating retrieval system.

4.1.2 Benefit and Drawback of TREC

Benefits :

1. Made research systems scale to large collections (pre-WWW).
2. Allows for somewhat controlled comparisons.

Drawbacks :

1. Emphasis on high recall, which may be unrealistic for what most users want.
2. Very long queries, also unrealistic.
3. Comparisons still difficult to make, because systems are quite different on many dimensions.
4. Focus on batch ranking rather than interaction.

4.2 User Happiness

- Google uses over 200 ranking signals, updates its algorithm over 500 times a year and employs thousands of engineers. Satisfaction is an actual ranking factor.
- User happiness is equated with the relevance of search results to the query. But how do you measure relevance ? Standard methodology in information retrieval consists of three elements.
  a) A benchmark document collection.
  b) A benchmark suite of queries.
  c) An assessment of the relevance of each query-document pair.
- User happiness can only be measured by relevance to an information need, not by relevance to queries.
- Issue : Who is the user we are trying to make happy ?
  1. Web engine : User finds what they want and return to the engine. Can we measure rate of return users ? User completes their task.

Scanned by TapScanner

2. eCommerce site : User finds what they want and buy. Is it the end-user or the eCommerce site, whose happiness we measure ? Measure time to purchase or fraction of searchers who become buyers ?
3. Recommender system : Users finds the recommendations useful OR the system is good at predicting the user rating ?

- Most common proxy : relevance of search results.
- Relevance measurement requires 3 elements :
  1. A benchmark document collection.
  2. A benchmark suite of queries.
  3. A usually binary assessment of either relevant or nonrelevant for each query and each document.

4.2.1 Search Engine Optimization

- Search Engine Optimization (SEO) is the act of modifying a website to increase its ranking in organic (vs paid), crawler-based listings of search engines. There are several ways to increase the visibility of your website through the major search engines on the Internet today. The two most common forms of Internet marketing Paid (Sponsored) Placement and Natural Placement. Both forms of optimization are acceptable marketing techniques, and each have advantages and disadvantages.
- Paid placement is a form of Search Engine Optimization (SEO) that was created by the major search engine providers such as Google, Yahoo, MSN, and AOL.
- Paid search engine placement has been an industry service that many companies have implemented into their strategy.
- The major search engine providers have not followed an ethical approach to their style of sponsored placement and have been accused of paying their employees to "click" on competing website sponsored links in order to increase the bidding war between competitors while boosting the revenue of the search engine companies.
- Consumers are skeptical of the paid placement marketing, but are often forced to use such techniques as paid placement in order to keep an edge over their competitors.

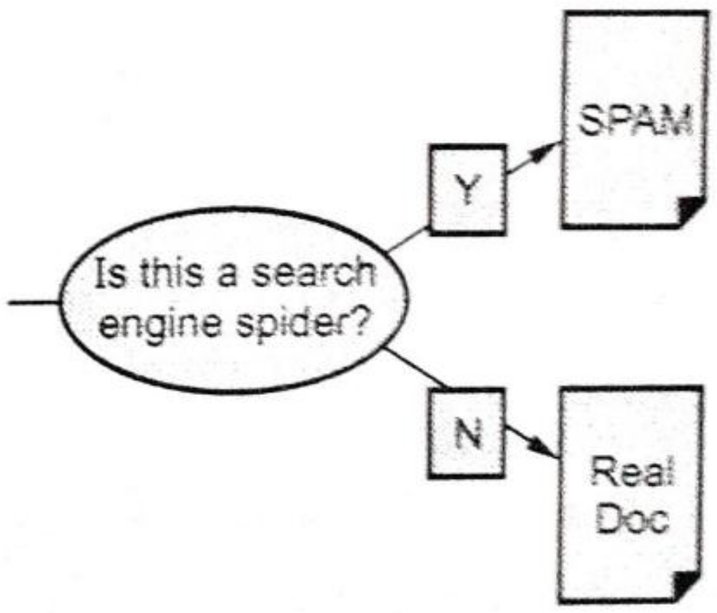


Fig. 4.2.1 SPAM

Scanned by TapScanner

- **Natural Placement** is a superior form of search engine marketing for many industries. While it may be increasingly difficult to maintain a high ranking in the search engines through natural placement, many companies are discovering that it is the strongest form of Internet marketing. Search engine programmers are creating extremely complex algorithms in order to decrease the amount of invalid optimization.

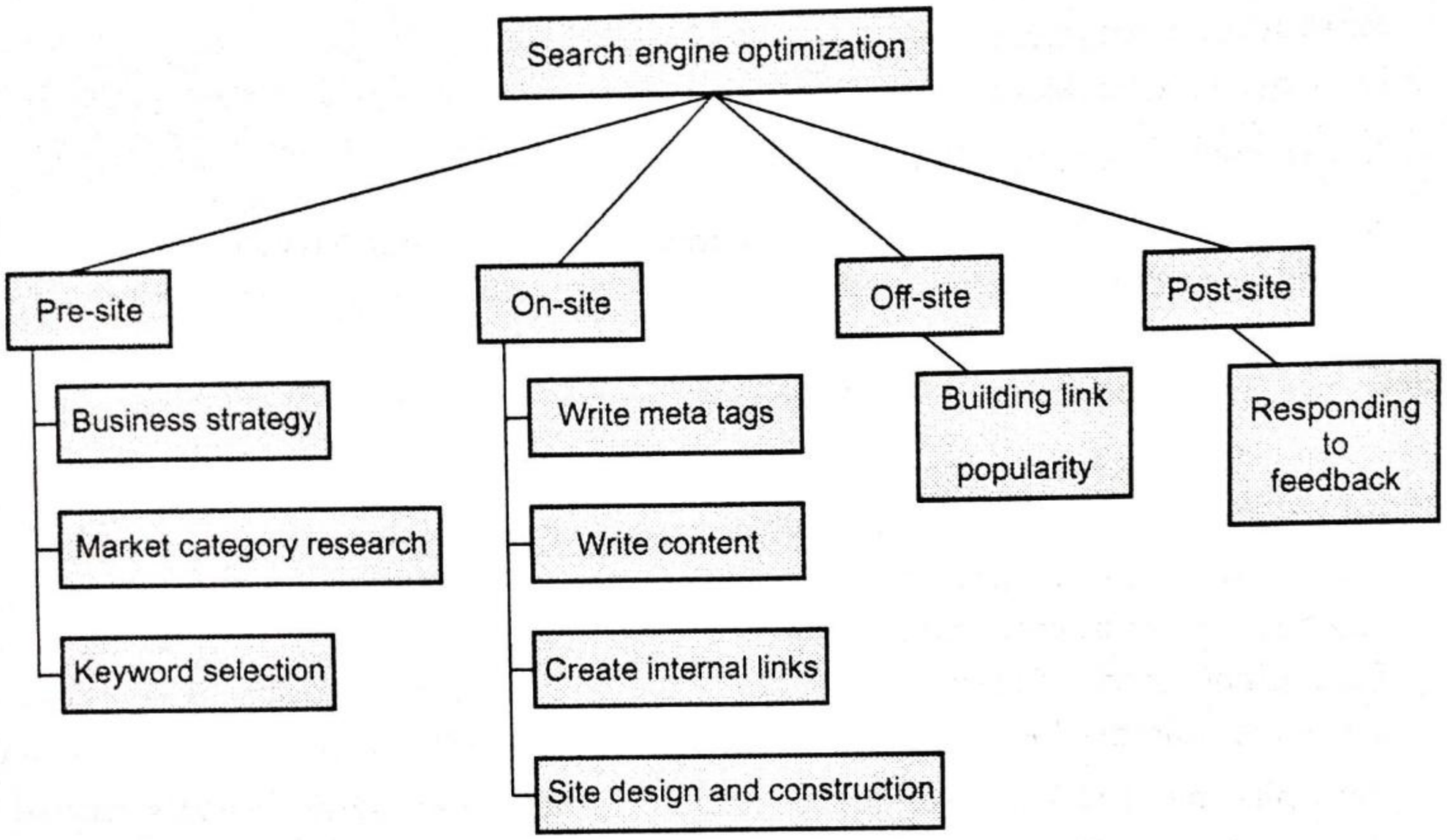


Fig. 4.2.2

- SEO refers to the process of "optimizing" both the on-page and off-page ranking factors in order to achieve high search engine rankings for targeted search terms.
- It also refers to the "industry" that revolves around obtaining high rankings in the search engines for desirable keyword search terms as a means of increasing the relevant traffic to a given website.
- SEO refers to an individual or company that optimizes websites for its clientele. It has a number of related meanings and usually refers to an individual/firm that focuses on optimizing for "organic" search engine rankings.
- **How do organic search listings work ?**
  1. A *spider* or *crawler* which is a component of a SE gathers listings by automatically "crawling" the web.
  2. The spider follows links to web pages, makes copies of the pages and stores them in the SE's index.
  3. Based on this data, the SE then *indexes* the pages and ranks the websites.
  4. Major SEs that index pages using spiders : Google, Altavista, Msn, Aol, lycos.

Scanned by TapScanner

SEO Techniques :

1. Domain name strategies
2. Linking strategies
3. Keywords
4. Title tags
5. Meta description tags
6. Alt tags
7. Submit your website to SEs for indexing

Benefit of SEO :

1. Increase your search engine visibility.
2. Generate more traffic from the major search engines.
3. Make sure your website and business get NOTICED and VISITED.
4. Grow your client base and increase business revenue.

4.2.2 Combining Searching and Browsing

- The two paradigms of searching and browsing are currently almost always used separately. Software tool called WebGlimpse that combines the two paradigms.
- Browsing and searching are the two main paradigms for finding information on line. The search paradigm has a long history; search facilities of different kinds are available in all computing environments. The browsing paradigm is newer and less ubiquitous, but it is gaining enormous popularity through the World-Wide Web.
- Both paradigms have their limitations. Search is sometimes hard for users who do not know how to form a search query so that it is limited to relevant information. Browsing can make the content come alive, and it is therefore more satisfying to users who get positive reinforcement as they proceed. However, browsing is time-consuming and users tend to get disoriented and lose their train of thoughts and their original goals.

WebGlimpse works as follows :

- At indexing time, it analyzes a given WWW archive, computes neighborhoods, adds search boxes to selected pages, collects remote pages when relevant and caches those pages locally.
- Once indexing is done, users who browse that site can search from any of the added search boxes and limit their search to the neighborhood of that page.

Scanned by TapScanner

- Glimpse's index consists essentially of two parts : The **word file** and the **pointers file.** The word file is simply a list of all the words in all documents, each followed by an offset into the pointers file. The pointers file contains for each word a list of pointers into the original text where that word appears.
- A search typically consists of two stages :
  1. First, the word file is searched and all relevant offsets into the pointers file are found. The relevant pointers in the pointers file are collected.
  2. The second stage is another search, using agrep, in the corresponding places in the original text. This is similar in principle to the usual inverted indexes, except that the word file, being one relatively small file, can be searched sequentially.
- This allows glimpse to support very flexible queries, including approximate matching, matching to parts of words and regular expressions. These flexible queries are implemented by running agrep directly on the word file. The fact that the files are searched directly allows the user to decide on-the-fly how much of the match to see.
- Glimpse's default is to show one line per match, but it can also show one paragraph or any user-defined record. This gives context to every match.

4.3 Creating Test Collections

- We can use precision and recall to measure the performance of IR system. Creating good test queries and useful document collections is hard. So we often build standard test collections.
- Test collection is
  1. A collection of documents.
  2. A set of information needs or queries.
  3. Relevance judgments.
- Examples : CRAN, CACM, TREC.

4.3.1 The Cranfield Collection

- This was the pioneering test collection in allowing precise quantitative measures of information retrieval effectiveness, but is nowadays too small for anything but the most elementary pilot experiments.
- Collected in the United Kingdom starting in the late 1950s, it contains 1,398 abstracts of aerodynamics journal articles, a set of 225 queries and exhaustive relevance judgments of all (query, document) pairs.

Scanned by TapScanner

- In the cranfield project, retrieval experiments were conducted on test databases in a controlled, laboratory like setting. The cranfield projects provided a foundation for the evaluation of IR systems.
- The objective was to study what kinds of indexing languages were most effective. At this time documents were indexed manually with a few keywords from controlled vocabularies/thesauri.

4.3.2 The CACM and ISI Collection

- The TREC collection is a large collection which requires time consuming preparation before experiments can be carried out effectively at a local site. A small collection might include features which are not present in the larger TREC collection.
- For the experimental studies in five different test collections were developed: ADI (documents on information science), CACM, INSPEC (abstracts on electronics, computer and physics), ISI, and Medlars (medical articles).

CACM collections

- It is small collections about computer science literature. It is text of 3,204 documents. The documents in the CACM test collection consist of all articles published in the communication of the ACM.
- CACM collection also includes information on structured subfields as follows :
  1. Word stems from the title and abstract sections.
  2. Categories.
  3. Direct references between articles.
  4. Bibliographic coupling connections.
  5. Number of co-citations for each pair of articles.
  6. Author names.
  7. Date information.
- A unique environment for testing retrieval algorithms which are based on information derived from cross-citing patterns. The CACM collection also includes a set of 52 test information requests.
- Example : "What articles exist which deal with TSS (Time Sharing System), an operating system for IBM computers ?"
- Also includes two Boolean query formulations and a set of relevant documents. Since the requests are fairly specific, the average number of relevant documents for each request is small (around 15). Precision and recall tend to be low.

Scanned by TapScanner

The ISI collection

- The 1,460 documents in the ISI test collection were selected from a previous collection assembled by Small at ISI (Institute of Scientific Information).
- The documents selected were those most cited in a cross-citation study done by Small. The main purpose is to support investigation of similarities based on terms and on cross-citation patterns.
- The documents in the ISI collection include three types of subfields as follows.
  1. Author names.
  2. Word stems from the title and abstract sections.
  3. Number of co-citations for each pair of articles.
- The ISI collection includes a total of 35 test information requests for which there are Boolean query formulations. It also includes 41 additional test information requests for which there is no Boolean query formulation.

Statistics for the CACM and ISI collections

1. Document statistics for the CACM and ISI Collections

Collection	Number of documents	Number of terms	Terms / Documents
CACM	3204	10446	40.1
ISI	1460	7392	104.9

2. Query statistics for the CACM and ISI Collections

Collection	Number of queries	Terms per query	Relevant per query	Relevant in top 10
CACM	52	11.4	15.3	1.9
ISI	35 and 76	8.1	49.8	1.7

4.3.3 Kappa Measure for Inter - Judge Agreement

- Kappa is a statistical measure of inter-rater reliability for categorical variables. Kappa is used when two raters both apply a criterion based on a tool to assess whether or not some condition occurs. Examples include :
  1. Two doctors rate whether or not each of 20 patients has diabetes based on symptoms.
  2. Two swallowing experts assess whether each of 30 stroke patients has aspirated while drinking based on their cough reaction.

Scanned by TapScanner

3. Two education researchers both assess whether each of 50 children reads proficiently based on a reading evaluation.

- If the two-raters can reliably use the criterion to make the same assessment on the same targets, then their agreement will be very high and provide evidence we have reliable ratings. If they can't, then either the criterion tool isn't useful or the raters are not well enough trained.
- Kappa measure
  a) Agreement measure among judges.
  b) Designed for categorical judgments.
  c) Corrects for chance agreement

$$\text{Kappa} = \frac{P(A) - P(E)}{1 - P(E)}$$

Where :

P(A) : Proportion of time judges agree.

P(E) : What agreement would be by chance.

Kappa = 0 for chance agreement, 1 for total agreement

- Kappa is always less than or equal to 1. A value of 1 implies perfect agreement and values less than 1 imply less than perfect agreement. In rare situations, kappa can be negative. This is a sign that the two observers agreed less than would be expected just by chance.

4.4 Multiple Choice Questions

Q.1 SEO stands for _____.

a Search Engine Optimization
b Search Engine Optimum
c Search Electronic Optimization
d None

Q.2 Which of the following are the TREC conferences uses for evaluations ?

a Summary table statistics
b Recall-precision average
c Document level averages
d All of these

Q.3 Which of the following items search engines don't want ?

a Keyword stuffing
b Buying links
c Poor user experience
d All of the above

Scanned by TapScanner

Q.4 Kappa is a statistical measure of inter-rater reliability for _____ variables.

a binary
b categorical
c static
d dynamic

Q.5 Kappa is a statistical measure of _____ reliability for categorical variables.

a inter-surveyor consistency
b intra-observer validity
c intra-coder validity
d inter-rater

Q.6 What is anchor text ?

a It is the main body of text on a particular web page.
b The text within the left or top panel of a web page.
c It is the visible text that is hyperlinked to another page.
d It is the most prominent text on the page that the search engines use to assign a title to the page.

Q.7 What is keyword density ?

a Number of times the keyword is used/total word count on page.
b The number of times the keyword used in title.
c The no of times the keyword used multiply by total no of word count on page.
d The number of times the keyword used in page description.

Q.8 A matrix containing correct and incorrect predictions in the form of TPs, FPs, FNs and TNs is known as _____.

a sparse matrix
b confusion matrix
c zero matrix
d all of these

Q.9 _____ occur when you predict an observation belongs to a class when in reality it does not.

a False negatives
b False positives
c True positives
d True negatives

Q.10 _____ occur when you predict an observation does not belong to a class when in fact it does.

a False negatives
b False positives
c True positives
d True negatives

Scanned by TapScanner

Answer Keys for Multiple Choice Questions

Q.1	a	Q.2	d	Q.3	d	Q.4	b
Q.5	d	Q.6	c	Q.7	a	Q.8	b
Q.9	b	Q.10	a				

❑❑❑

Scanned by TapScanner

5 Text Categorization and Filtering

Syllabus

Introduction to text classification. Naive Bayes models. Spam filtering. Vector space classification using hyperplanes; centroids; k Nearest Neighbors. Support vector machine classifiers. Kernel functions. Boosting.

Contents

5.1 *Introduction to Text Classification*

5.2 *Naive Bayes Models*

5.3 *k-Nearest Neighbors*

5.4 *Support Vector Machine Classifiers*

5.5 *Boosting*

5.6 *Multiple Choice Questions*

Scanned by TapScanner

5.1 Introduction to Text Classification

Text classification is the process of assigning tags or categories to text according to its content.

5.1.1 Machine Learning

- A (machine learning) problem is well-posed if a solution to it exists, if that solution is unique, and if that solution depends on the data / experience but it is not sensitive to (reasonably small) changes in the data / experience.
- The main goal of machine learning is to study, engineer, and improve mathematical models which can be trained with context-related data to infer the future and to make decisions without complete knowledge of all influencing elements.
- We discuss basic design issues and approaches to machine learning.
- Goal : Design a system to learn how to play checkers and enter it into the world checkers tournament.
  1) Choose the training experience
  2) Choose the target function
  3) Choose a representation for the target function
  4) Choose a function approximation algorithm
- Main goal of machine learning is to devise learning algorithms that do the learning automatically without human intervention or assistance. The machine learning paradigm can be viewed as "programming by example." Another goal is to develop computational models of human learning process and perform computer simulations.
- The goal of machine learning is to build computer systems that can adapt and learn from their experience.
- Algorithm is used to solve a problem on computer. An algorithm is a sequence of instruction. It should carry out to transform the input to output.
- For example, for addition of four numbers is carried out by giving four number as input to the algorithm and output is sum of all four numbers.
- For the same task, there may be various algorithms. It is interested to find the most efficient one, requiring the least number of instructions or memory or both.

Why is Machine Learning Important ?

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.

Scanned by TapScanner

- Machine learning provides business insight and intelligence. Decision makers are provided with greater insights into their organizations. This adaptive technology is being used by global enterprises to gain a competitive edge.
- Machine learning algorithms discover the relationships between the variables of a system (input, output and hidden) from direct samples of the system.

Supervised Learning :

- Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples.
- The task of the supervised learner is to predict the output behavior of a system for any set of input values, after an initial training phase.
- Supervised learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs are usually provided by an external teacher.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or a regression function.

Un-Supervised Learning :

- The model is not provided with the correct results during the training. It can be used to cluster the input data in classes on the basis of their statistical properties only. Cluster significance and labelling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes. All similar inputs patterns are grouped together as clusters.
- If matching pattern is not found, a new cluster is formed. There is no error feedback.
- They are called unsupervised because they do not need a teacher or super-visor to label a set of training examples. Only the original data is required to start the analysis.

Semi-Supervised Learning

- Semi-supervised learning uses both labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone.
- Semi-supervised learning is motivated by its practical value in learning faster, better, and cheaper. In many real-world applications, it is relatively easy to acquire a large amount of unlabeled data x.
- For example, documents can be crawled from the Web, images can be obtained from surveillance cameras, and speech can be collected from broadcast.

Scanned by TapScanner

- Semi-supervised Learning makes use of both labeled and unlabeled data for training, typically a small amount of labeleddata with a large amount of unlabeled data.

5.1.2 Text Classification Problem

- "In text classification, given a description d ∈ X of a document, where x is the document space ; and a fixed set of classes C = {c_1, c_2, c_3 ..... c_j}. Classes are also called *categories* or *labels*.
- "Typically, the document space X is some type of high-dimensional space, and the classes are human defined for the needs of an application, as in the examples India and documents that talk about multicore computer chips.
- For given a *training set* D of labeled documents < d, c > where < d, c > ∈ X × C.
- For example : < d, c > = < Mumbai joins the World Trade Organization, India > for the one-sentence document Mumbai joins the World Trade Organization and the class (or label) India.
- If no restrictions are posed to the classifier, two or more classes might be assigned to a single document.

5.1.3 Text Classification Algorithm

- Text classification algorithms are classified as supervised or unsupervised algorithm. Unsupervised algorithm are used for large collections and without training data. Supervised algorithm requires training data.
- Supervised text classification algorithms are decision tree, nearest neighbours, relevance feedback, naïve Bayes, support vector machine, ensemble etc.
- Unsupervised algorithms are clustering and Naïve text classification.

5.2 Naive Bayes Models

- Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. It is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.
- A Naive Bayes classifier is a program which predicts a class value given a set of attributes.
- For each known class value,
  1. Calculate probabilities for each attribute, conditional probability for the attributes.

Scanned by TapScanner

2. Use the product rule to obtain a joint conditional probability for the attributes.
3. Use Bayes rule to derive conditional probabilities for the class variable.

- Once this has been done for all class values, output the class with the highest probability.
- Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class.

5.2.1 Spam Filtering

- Spam filtering is the best-known use of Naïve Bayesian text classification.
- Presently, almost all the email providers have this as a built-in functionality, which makes use of a Naïve Bayes classifier to identify spam email on the basis of certain conditions and also the probability of classifying an email as 'Spam'.
- Naïve Bayesian spam sifting has turned into a mainstream mechanism to recognize illegitimate a spam email from an honest-to-goodness email.
- Users can also install separate email filtering programmes. Server-side email filters such as DSPAM, Spam Assassin, Spam Bayes, and ASSP make use of Bayesian spam filtering techniques, and the functionality is sometimes embedded within the mail server software itself.
- Once the data is given, a filter is created that assigns the probability that each feature is in spam. Probabilities in this case are written as values between 0 and 1. Common words such as "the" and "it" will be assigned very neutral probabilities (around 0.5).
- In many cases, these words will simply be ignored by the filter, as they offer very little relevant information.
- Now that the filter is ready to detect spam from the training data, it will evaluate whether or not an email is spam based on the individual probabilities of each word.
- Fig. 5.2.1 shows that the words are independent from each other in the classification of the email, and are all treated equally.

Scanned by TapScanner

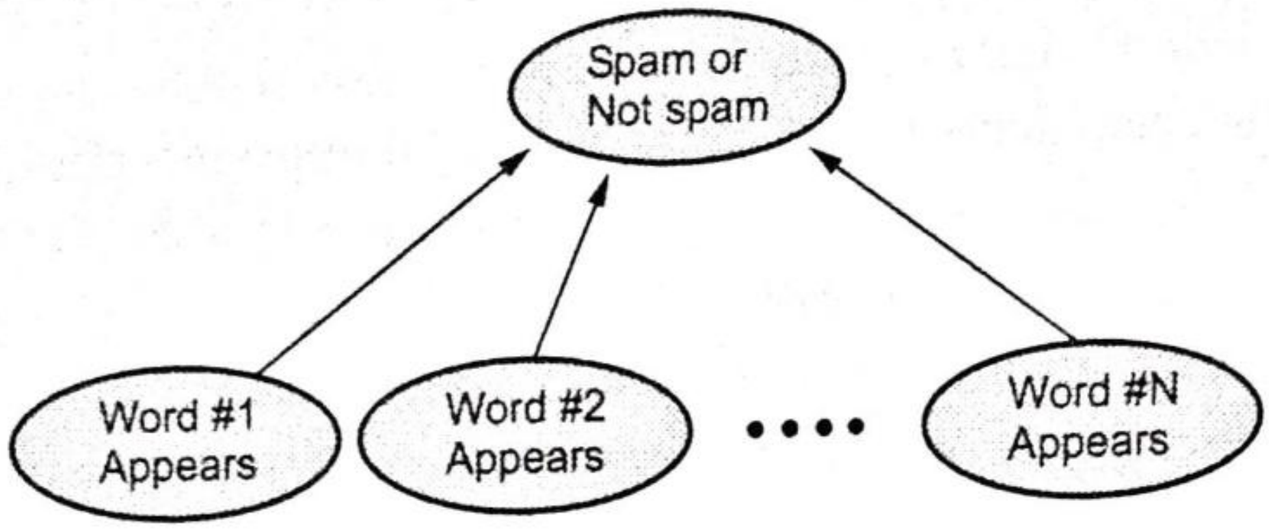


Fig. 5.2.1

- Formula for P(S|W), the probability that an email is spam given that word W is in the email. This is also called the spamicity of a word

$$P(S|W) = \frac{P(W|S) \cdot P(S)}{P(W|S) \cdot P(S) + P(W|H) \cdot P(H)}$$

- W represents a given word, S represents a spam email and H represents a ham email. The left side of the equation can be read as "The probability that an email is spam given that the email contains word W."
- P(W|S) and P(W|H) are the conditional probabilities of the word W. For P(W|S), this would be the proportion of spam documents that also contain word W. P(W|H), similarly, is the proportion of ham documents that also contain word W.
- These probabilities come from the training data. P(S) the probability that any given email is spam and P(H) the probability that any given email is ham vary depending on the approach.
- Another approach is to use the data from the training set used to train the filter, which makes the filter more tailored for the user since the user classified the training data.

5.3 k-Nearest Neighbors

- The K-nearest neighbor (KNN) is a classical classification method and requires no training effort, critically depends on the quality of the distance measures among examples.
- The KNN classifier uses Mahalanobis distance function. A sample is classified according to the majority vote of its nearest K training samples in the feature space. Distance of a sample to its neighbors is defined using a distance function.

Scanned by TapScanner

- For all points x, y and z, a distance function F(. , .), must satisfy the following conditions :

1.	Nonnegativity	$F(x, y) \geq 0$
2.	Reflexivity	$F(x, y) = 0$ if and only if $x = y$
3.	Symmetry	$F(x, y) = F(y, x)$
4.	Triangle inequality	$F(x, y) + F(y, x) \geq F(x, y)$

The Euclidean distance

- The Euclidean distance is the most common distance metric used in low dimensional data sets. It is also known as the **L_2 norm.** The Euclidean distance is the usual manner in which distance is measured in real world.

$$d_{euclidean}(x, y) = \sqrt{\sum i (x_i - y_i)^2}$$

where x and y are m-dimensional vectors and denoted by $x = \{x_1, x_2, x_3, \ldots, x_m\}$ and $y = \{y_1, y_2, y_3, \ldots, y_m\}$ represent the m attribute values of two records.

- While Euclidean metric is useful in low dimensions, it doesn't work well in high dimensions and for categorical variables. The drawback of Euclidean distance is that it ignores the similarity between attributes. Each attribute is treated as totally different from all of the attributes.

Mahalanobis distance

- Mahalanobis distance is also called quadratic distance.
- Mahalanobis distance is a distance measure between two points in the space defined by two or more correlated variables. Mahalanobis distance takes the correlations within a data set between the variableinto considerations.
- If there are two non-correlated variables, the Mahalanobis distance between the points of the variable in a 2D scatter plot is same as Euclidean distance.
- The Mahalanobis distance is the distance between an observation and the center for each group in m-dimensional space defined by m variables and their covariance. Thus, a small value of Mahalanobis distance increases the chance of an observation to be closer to the group's center and the more likely it is to be assigned to that group.
- Mahalanobis distance between two samples (x, y) of a random variable is defined as

$$d_{Mahalanobis}(x, y) = \sqrt{(x-y)^T \sum^{-1} (x-y)}$$

- The Mahalanobis metric is defined in independence of the data matrix.

Scanned by TapScanner

- No pre-processing of labeled data samples is needed before using KNN algorithm. A dominated class label in K-nearest neighbors of a data point is assigned as class label to that data point. A tie occurs when neighborhood has same amount of labels from multiple classes.
- To break the tie, the distances of neighbors can be summed up in each class that is tied and vector f is assigned to the class with minimal distance. Or, the class can be chosen with the nearest neighbor. Clearly, tie is still possible here, in which case an arbitrary assignment is taken.
- Three distance functions that can be used in KNN classifier are :

1	L_p norm	$L_p(x, y) = \left(\sum_{i=1}^{d} \lvert x_i - y_i \rvert^p\right)^{1/p}$
2	L_2 norm (Euclidean distance)	$L_2(x, y) = \left(\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2\right)^{1/2}$
3	L_1 norm (Manhattan distance)	$L_1(x, y) = \sum_{i=1}^{d} \lvert x_i - y_i \rvert$

- Mahalanobis distance that takes into account the correlation S of the dataset :

$$L_m(x, y) = \sqrt{(x-y)\, S^{-1} (x-y)}$$

- The steps that need to be carried out during the KNN algorithm are as follows :
  a. Divide the data into training and test data.
  b. Select a value K.
  c. Determine which distance function is to be used.
  d. Choose a sample from the test data that needs to be classified and compute the distance to its n training samples.
  e. Sort the distances obtained and take the k-nearest data samples.
  f. Assign the test class to the class based on the majority vote of its K neighbors.
- Fig. 5.3.1 shows geometrical representation of K-nearest neighbor algorithm.

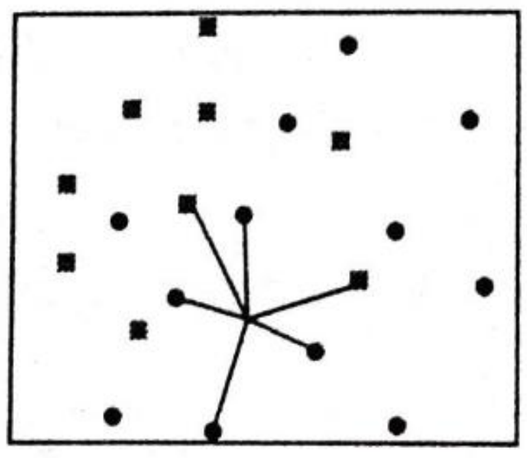


Fig. 5.3.1

Scanned by TapScanner

- The performance of the KNN algorithm is influenced by three main factors :
  1. The distance function or distance metric used to determine the nearest neighbors.
  2. The decision rule used to derive a classification from the K-nearest neighbors.
  3. The number of neighbors used to classify the new example.

Advantages

1. The KNN algorithm is very easy to implement.
2. Nearly optimal in the large sample limit.
3. Uses local information, which can yield highly adaptive behavior.
4. Lends itself very easily to parallel implementations.

Disadvantages

1. Large storage requirements.
2. Computationally intensive recall.
3. Highly susceptible to the curse of dimensionality.

5.4 Support Vector Machine Classifiers

- Support Vector Machines (SVMs) are a set of supervised learning methods which learn from the dataset and used for classification. SVM is a classifier derived from statistical learning theory by Vapnik and Chervonenkis.
- An SVM is a kind of large-margin classifier : it is a vector space based machine learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data.
- Given a set of training examples, each marked as belonging to one of two classes, an SVM algorithm builds a model that predicts whether a new example falls into one class or the other. Simply speaking, we can think of an SVM model as representing the examples of the separate classes are divided by a gap that is as wide as possible.
- New examples are then mapped into the same space and classified to belong to the class based on which side of the gap they fall on.
- SVM are primarily two-class classifiers with the distinct characteristic that they aim to find the optimal hyperplane such that the expected generalization error is minimized. Instead of directly minimizing the empirical risk calculated from the training data, SVMs perform structural risk minimization to achieve good generalization.

Scanned by TapScanner

- The empirical risk is the average loss of an estimator for a finite set of data drawn from P. The idea of risk minimization is not only measure the performance of an estimator by its risk, but to actually search for the estimator that minimizes risk over distribution P. Because we don't know distribution P we instead minimize empirical risk over a training dataset drawn from P. This general learning technique is called **empirical risk minimization.**
- Fig. 5.4.1 shows empirical risk.

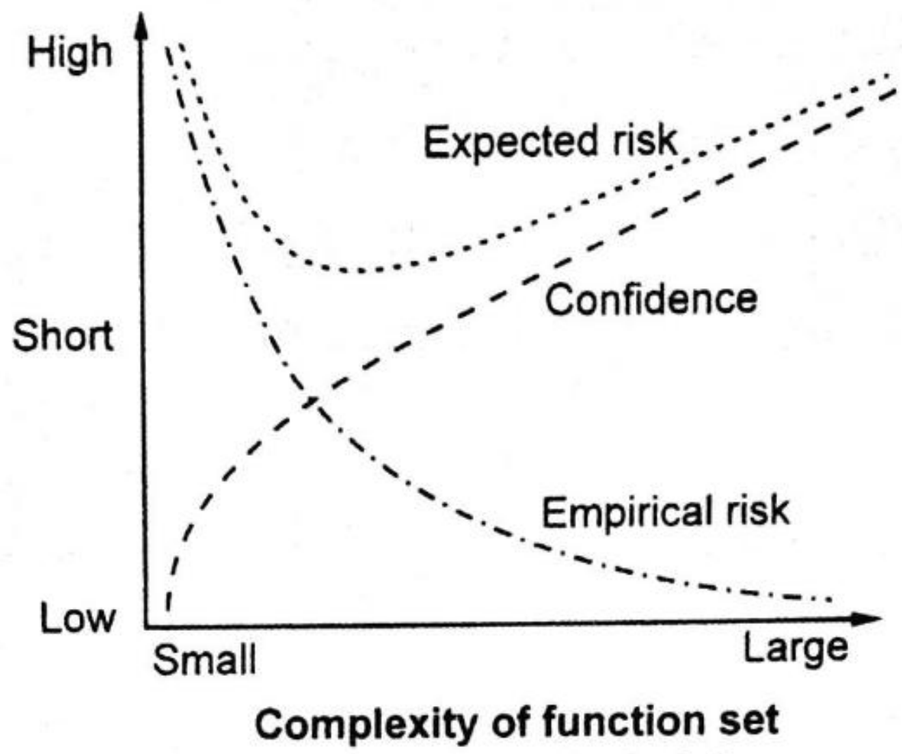


Fig. 5.4.1 : Empirical risk

Key Properties of Support Vector Machines

1. Use a single hyperplane which subdivides the space into two half-spaces, one which is occupied by Class 1 and the other by Class 2.
2. They maximize the margin of the decision boundary using quadratic optimization techniques which find the optimal hyperplane.
3. Ability to handle large feature spaces.
4. Overfitting can be controlled by soft margin approach.
5. When used in practice, SVM approaches frequently map the examples to a higher dimensional space and find margin maximal hyperplanes in the mapped space, obtaining decision boundaries which are not hyperplanes in the original space.
6. The most popular versions of SVMs use non-linear kernel functions and map the attribute space into a higher dimensional space to facilitate finding "good" linear decision boundaries in the modified space.

Soft Margin SVM

- For the very high dimensional problems common in text classification, sometimes the data are linearly separable. But in the general case they are not, and even if they are, we might prefer a solution that better separates the bulk of the data while ignoring a few weird noise documents.

Scanned by TapScanner

- What if the training set is not linearly separable ? *Slack variables* can be added to allow misclassification of difficult or noisy examples, resulting margin called *soft*.
- A *soft-margin* allows a few variables to cross into the margin or over the hyperplane, allowing misclassification.
- We penalize the crossover by looking at the number and distance of the misclassifications. This is a trade-off between the hyperplane violations and the margin size. The slack variables are bounded by some set cost. The farther theyare from the soft margin, the less influence they have on the prediction.
- All observations have an associated slack variable.
  1. Slack variable = 0 then all points on the margin.
  2. Slack variable > 0 then a point in the margin or on the wrong side of the hyperplane.
  3. C is the tradeoff between the slack variable penalty and the margin.

Limitations of SVM

1. It is sensitive to noise.
2. The biggest limitation of SVM lies in the choice of the kernel.
3. Another limitation is speed and size.
4. The optimal design for multiclass SVM classifiers is also a research area.

5.4.1 Kernel Functions

- SVM and number of other linear classifiers, provide an easy and efficient way of doing mapping to a higher dimensional space, which is referred to as the kernel trick. The SVM linear classifier relies on a dot product between data point vectors.
  - If every datapoint is mapped into high - dimensional space via some transformation $\phi : x \rightarrow \phi(x)$, the inner product becomes :

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

  - A kernel function is some function that corresponds to an inner product in some expanded feature space.
- Kernel functions are sometimes more precisely referred to as *Mercer kernels*, because they must satisfy Mercer's condition.
- A kernel function K must be continuous, symmetric, and have a positive definite gram matrix. Such a K means that there exists a mapping to a reproducing kernel Hilbert space such that the dot product there gives the same value as the function K.

Scanned by TapScanner

- If a kernel does not satisfy Mercer's condition, then the corresponding QP may have no solution.
- The two commonly used families of kernels are polynomial kernels and radial basis functions. The most common form of radial basis function is a Gaussian distribution.
- A radial basis function is equivalent to mapping the data into an infinite dimensional Hilbert space, and so we cannot illustrate the radial basis function concretely, as we did a quadratic kernel.

5.5 Boosting

- Boosting is a very different method to generate multiple predictions (function estimates) and combine them linearly. Boosting refers to a general and provably effective method of producing a very accurate classifier by combining rough and moderately inaccurate rules of thumb.
- Originally developed by computational learning theorists to guarantee performance improvements on fitting training data for a ***weak learner*** that only needs to generate a hypothesis with a training accuracy greater than 0.5. Final result is the weighted sum of the results of weak classifiers.
- A learner is weak if it produces a classifier that is only slightly better than random guessing, while a learner is said to be strong if it produces a classifier that achieves a low error with high confidence for a given concept.
- Revised to be a practical algorithm, AdaBoost, for building ensembles that empirically improves generalization performance. Examples are given weights. At each iteration, a new hypothesis is learned and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.
- Boosting is a bias reduction technique. It typically improves the performance of a single tree model. A reason for this is that we often cannot construct trees which are sufficiently large due to thinning out of observations in the terminal nodes.
- Boosting is then a device to come up with a more complex solution by taking linear combination of trees. In presence of high-dimensional predictors, boosting is also very useful as a regularization technique for additive or interaction modeling.
- To begin, we define an algorithm for finding the rules of thumb, which we call a weak learner. The boosting algorithm repeatedly calls this weak learner, each time feeding it a different distribution over the training data. Each call generates a weak classifier and we must combine all of these into a single classifier that, hopefully, is much more accurate than any one of the rules.

Scanned by TapScanner

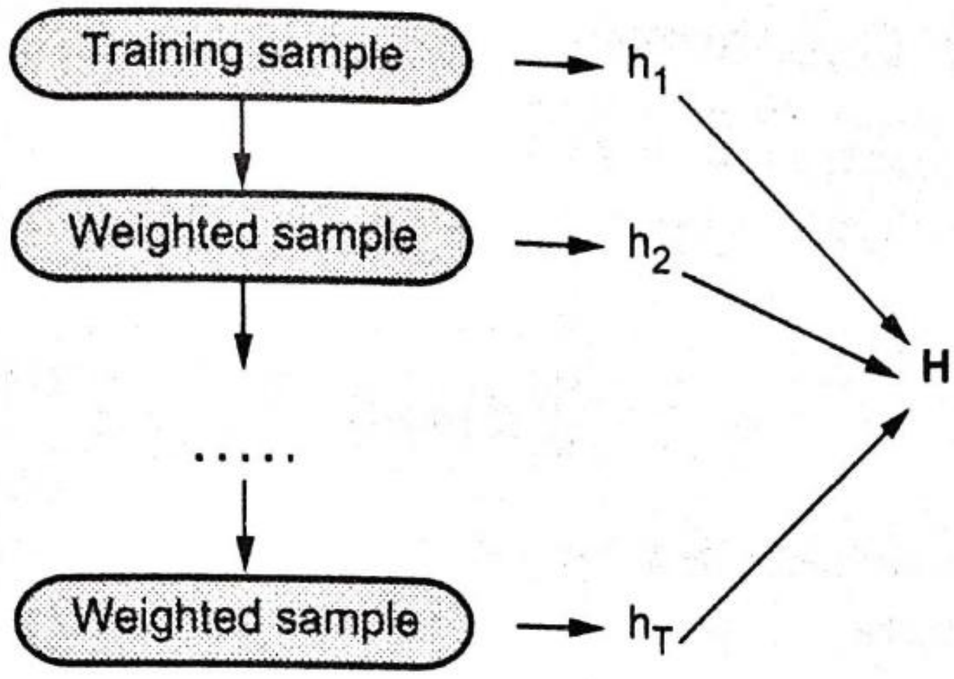


Fig. 5.5.1

- Train a set of weak hypotheses : $h_1, ..., h_T$. The combined hypothesis H is a weighted majority vote of the T weak hypotheses. During the training, focus on the examples that are misclassified.

AdaBoost :

- AdaBoost, short for "Adaptive Boosting", is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire who won the prestigious "Gödel Prize" in 2003 for their work. It can be used in conjunction with many other types of learning algorithms to improve their performance.
- It can be used to learn weak classifiers and final classification based on weighted vote of weak classifiers.
- It is linear classifier with all its desirable properties. It has good generalization properties.
- To use the weak learner to form a highly accurate prediction rule by calling the weak learner repeatedly on different distributions over the training examples.
- Initially, all weights are set equally, but each round the weights of incorrectly classified examples are increased so that those observations that the previously classifier poorly predicts receive greater weight on the next iteration.

Advantages of AdaBoost :

1. Very simple to implement
2. Fairly good generalization
3. The prior error need not be known ahead of time.

Disadvantages of AdaBoost :

1. Suboptimal solution
2. Can over fit in presence of noise.

Scanned by TapScanner

5.6 Multiple Choice Questions

Q.1 **_____ can be cast as a problem of searching through a large predefined space of potential hypotheses.**

a Machine learning
b Concept learning
c Training set
d Find-S

Q.2 **A _____ is a flowchart - like tree structure, where each internal node denotes a test on an attribute.**

a decision tree
b binary tree
c cluster
d none of these

Q.3 **CART stands for _____.**

a Concept and Regression Trees
b Classification and Regression Trees
c Cluster and Regression Trees
d None

Q.4 **In theory, Bayesian classifiers have the _____ error rate in comparison to all other classifiers.**

a equal
b maximum
c minimum
d zero

Q.5 **Error rate = 1 – (---------- rate)**

a Recall
b Precision
c Specificity
d Accuracy

Q.6 **_____ classifiers are statistical classifiers.**

a Bayesian
b CART
c Information gain
d All of these

Q.7 **_____ is a technique used to predict group membership for data instances.**

a Clustering
b Steaming
c Classification
d None

Scanned by TapScanner

Q.8 _____ matrix is also called a contingency table.

a Plain
b Diffusion
c Performance
d Confusion

Q.9 Spam filtering is the best-known use of _____ text classification.

a Naïve Bayesian
b kNN
c Bayesian
d None

Q.10 Some of the popular classification models include _____.

a k-Nearest Neighbor
b Naïve Bayes
c Decision Tree
d All of these

Answer Keys for Multiple Choice Questions

Q.1	b	Q.2	a	Q.3	b	Q.4	c
Q.5	d	Q.6	a	Q.7	c	Q.8	d
Q.9	a	Q.10	d				

❑❑❑

Scanned by TapScanner

6 Text Clustering

Syllabus

Clustering versus classification. Partitioning methods. k-means clustering. Mixture of Gaussians model. Hierarchical agglomerative clustering. Clustering terms using documents.

Contents

6.1 *Clustering*

6.2 *Partitioning Methods*

6.3 *Mixture of Gaussians Model*

6.4 *Hierarchical Agglomerative Clustering*

6.5 *Clustering Terms using Documents*

6.6 *Multiple Choice Questions*

Scanned by TapScanner

6.1 Clustering

- A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.
- Conceptual clustering : Two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.
- Clustering methods are unsupervised learning techniques. cluster distance is a distance of two closest members in each class. Clustering methods are usually categorized according to the type of cluster they produce. The clustering methods are categorized as :
  1. Hierarchical methods : These types of cluster produces the output list of cluster. Small clusters of highly similar documents nested within larger clusters of less similar documents.
  2. Non-hierarchical methods : This method produced unordered lists.
- Other clustering methods are exclusive cluster and overlapping cluster. In the first case (exclusive cluster) data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster.
- The overlapping clustering uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.
- Cluster analysis involves applying one or more clustering algorithms with the goal of finding hidden patterns or groupings in a dataset. Clustering algorithms form groupings or clusters in such a way that data within a cluster have a higher measure of similarity than data in any other cluster. The measure of similarity on which the clusters are modeled can be defined by Euclidean distance, probabilistic distance, or another metric.
- Cluster analysis is an unsupervised learning method and an important task in exploratory data analysis. Popular clustering algorithms include :
  1. Hierarchical clustering : Builds a multilevel hierarchy of clusters by creating a cluster tree.
  2. k-Means clustering : Partitions data into k distinct clusters based on distance to the centroid of a cluster.
  3. Gaussian mixture models : Models clusters as a mixture of multivariate normal density components.

Scanned by TapScanner

4. Self-organizing maps : Uses neural networks that learn the topology and distribution of the data.

Desirable Properties of a Clustering Algorithm

1. Scalability (in terms of both time and space)
2. Ability to deal with different data types
3. Minimal requirements for domain knowledge to determine input parameters
4. Interpretability and usability

Distance between Clusters :

1. Single Link : Smallest distance between points
2. Complete Link : Largest distance between points
3. Average Link : Average distance between points
4. Centroid : Distance between centroids

Supervised learning after clustering

- Supervised clustering is the task of automatically adapting a clustering algorithm with the aid of a training set consisting of item sets and complete partitioning of these item sets.
- Dimensionality reduction methods find correlations between features and group features. Clustering methods find similarities between instances and group instances.
- It allows knowledge extraction through number of clusters, prior probabilities and cluster parameters, i.e., center, range of features.
- Example : CRM, customer segmentation
- One advantage of preceding a supervised learner with unsupervised clustering or dimensionality reduction is that the latter does not need labeled data. Labeling the data is costly. We can use a large amount of unlabeled data for learning the cluster parameters and then use a smaller labeled data to learn the second stage of classification or regression.

Examples of Clustering Applications

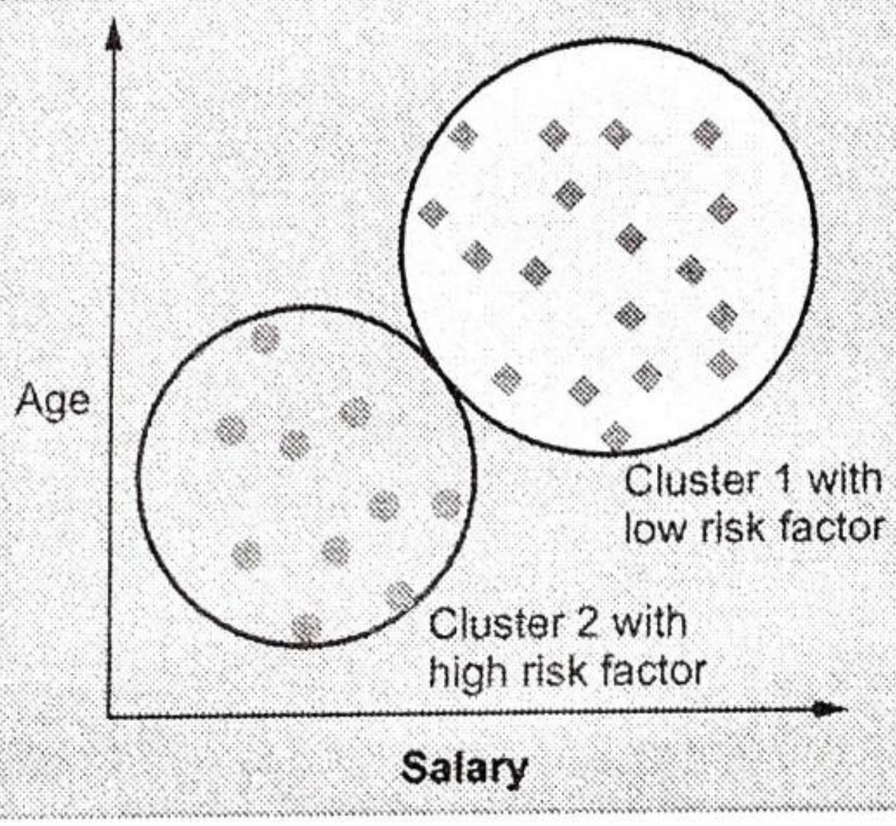
- **Marketing** : Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.
- **Land use** : Identification of areas of similar land use in an earth observation database.
- **Insurance** : Identifying groups of motor insurance policy holders with a high average claim cost.

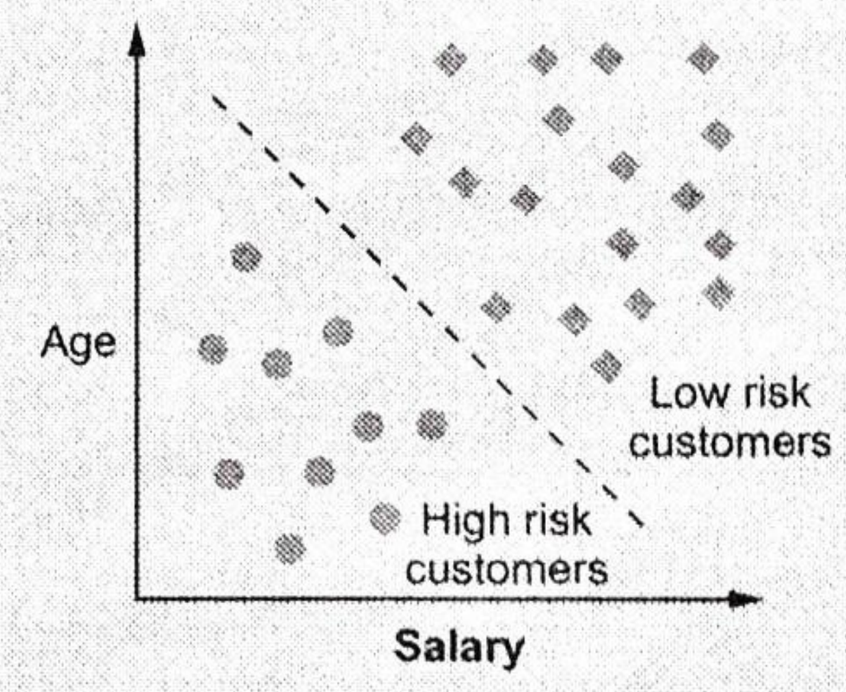
Scanned by TapScanner

- **City-planning :** Identifying groups of houses according to their house type, value, and geographical location.
- **Earthquake studies :** Observed earthquake epicenters should be clustered along continent faults.

6.1.1 Clustering Versus Classification

Clustering	Classification
Clustering is used to group similar data instances together.	Classification is used to label data.
No training data is required.	Training data is required.
Clustering uses unsupervised learning techniques.	Classification uses supervised learning techniques.
Uses unlabelled data as the input.	Uses labelled data as the input.
Clustering Algorithms : K-means, K-medoids, Density Based and Hierarchical.	Classification Algorithms : Logistic Regression, Decision Tree, Neural Networks, Naive Bayes, kNN and SVM.
Common Applications : Recommender System, Customer Segmentation.	Common Applications : Risk Assessment Model, Spam Detection and Fraud Detection.
Dataset consists of attributes.	Datasets consist of attributes and a class labels.





6.2 Partitioning Methods

- Partitioning clustering are clustering methods used to classify observations, within a data set, into multiple groups based on their similarity. The algorithms require the analyst to specify the number of clusters to be generated.
- Commonly used partitioning methods are k-means, k-medoids.

Scanned by TapScanner

6.2.1 K-Mean Clustering

- K-Means clustering is heuristic method. Here each cluster is represented by the center of the cluster. "K" stands for number of clusters, it is typically a user input to the algorithm; some criteria can be used to automatically estimate K.
- This method initially takes the number of components of the population equal to the final required number of clusters. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart.
- Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroid's position is recalculated everytime and component is added to the cluster. This continues until all the components are grouped into the final required number of clusters.
- Given K, the K-means algorithm consists of four steps :
  1. Select initial centroids at random.
  2. Assign each object to the cluster with the nearest centroid.
  3. Compute each centroid as the mean of the objects assigned to it.
  4. Repeat previous 2 steps until no change.
- The x_1, ..., x_N are data points or vectors of observations. Each observation (vector x_i) will be assigned to one and only one cluster. The C(i) denotes cluster number for the observation. K-means minimizes within-cluster point scatter :

$$W(C) = \frac{1}{2} \sum_{K-1}^{K} \sum_{C(i)-K} \sum_{C(i)-K} ||x_i - x_j||^2$$

$$= \sum_{K-1}^{K} N_K \sum_{C(i)-K} ||x_i - m_K||^2$$

where

m_K is the mean vector of the Kth cluster.

N_K is the number of observations in K^{th} cluster.

K-Means Algorithm Properties

1. There are always K clusters.
2. There is always at least one item in each cluster.
3. The clusters are non-hierarchical and they do not overlap.
4. Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'center' of clusters.

Scanned by TapScanner

The K-Means Algorithm Process

1. The dataset is partitioned into K clusters and the data points are randomly assigned to the clusters resulting in clusters that have roughly the same number of data points.
2. For each data point.
   a. Calculate the distance from the data point to each cluster.
   b. If the data point is closest to its own cluster, leave it where it is.
   c. If the data point is not closest to its own cluster, move it into the closest cluster.
3. Repeat the above step until a complete pass through all the data points results in no data point moving from one cluster to another. At this point the clusters are stable and the clustering process ends.
4. The choice of initial partition can greatly affect the final clusters that result, in terms of inter-cluster and intracluster distances and cohesion.

- K-means algorithm is iterative in nature. It converges, however only a local minimum is obtained. It works only for numerical data. This method easy to implement.

- **Advantages of K-Means Algorithm :**
  1. Efficient in computation
  2. Easy to implement
- **Weaknesses**
  1. Applicable only when mean is defined.
  2. Need to specify K, the number of clusters, in advance.
  3. Trouble with noisy data and outliers.
  4. Not suitable to discover clusters with non-convex shapes.

6.2.2 K-Medoids

- The K-medoids algorithm is a clustering algorithm related to the K-means algorithm and the medoidshift algorithm. K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into K clusters known a priori. A useful tool for determining K is the silhouette.
- The most common realisation of K-medoid clustering is the Partitioning Around Medoids (PAM) algorithm. PAM uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search.
- Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.

Scanned by TapScanner

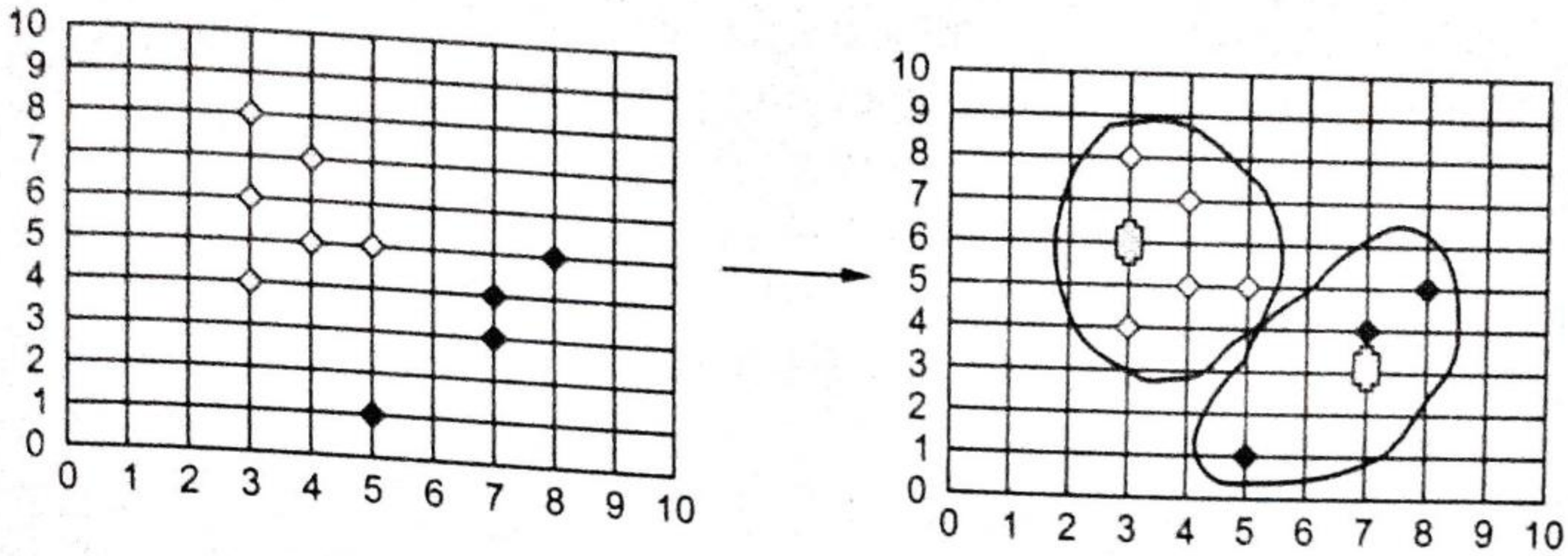


Fig. 6.2.1

- A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.
  1. The algorithm begins with arbitrary selection of the K objects as medoid points out of n data points (n > K).
  2. After selection of the k medoid points, associate each data object in the given data set to most similar medoid. The similarity here is defined using distance measure that can be euclidean distance, manhattan distance or minkowski distance.
  3. Randomly select nonmedoid object O'.
  4. Compute total cost, S of swapping initial medoid object to O'.
  5. If S < 0, then swap initial medoid with the new one (if S < 0 then there will be new set of medoids).
  6. Repeat steps 2 to 5 until there is no change in the medoid.

6.2.3 Silhouette

- Silhouette refers to a method of interpretation and validation of clusters of data.
- Silhouettes are a general graphical aid for interpretation and validation of cluster analysis. This technique is available through the silhouette function. In order to calculate silhouettes, two types of data are needed :
  1. The collection of all distances between objects. These distances are obtained from application of dist function on the coordinates of the elements in material with argument method.
  2. The partition obtained by the application of a clustering technique.
- For each element, a silhouette value is calculated and evaluates the degree of confidence in the assignment of the element :
  1. Well-clustered elements have a score near 1.

Scanned by TapScanner

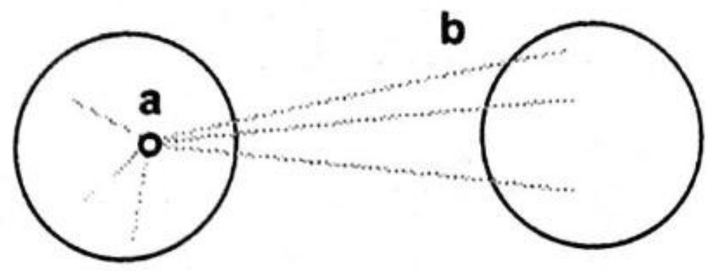
2. Poorly-clustered elements have a score near - 1.

- Thus, silhouettes indicate the objects that are well or poorly clustered. Silhouette coefficient combines ideas of both cohesion and separation, but for individual points, as well as clusters and clustering's.
- For an individual point, I

  a = Average distance of i to the points in the same cluster

  b = min (average distance of i to points in another cluster)

s = 1 – a/b if a < b



Silhouette coefficient of i :

Silhouette coefficient

- **Cohesion** : Measures how closely related are objects in a cluster.
- **Separation** : Measure how distinct or well-separated a cluster is from other clusters.

6.3 Mixture of Gaussians Model

- Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.
- A Gaussian mixture is a function that is comprised of several Gaussians, each identified by $k \in \{1,\ldots, K\}$, where K is the number of clusters of our dataset.
- Each Gaussian k in the mixture is comprised of the following parameters :
  a) A mean μ that defines its centre.
  b) A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
  c) A mixing probability π that defines how big or small the Gaussian function will be.
- Gaussian mixture model attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset.
- The main difficulty in learning Gaussian mixture models from unlabeled data is that it is one usually doesn't know which points came from which latent component.
- Expectation-maximization is a well-founded statistical algorithm to get around this problem by an iterative process.

Scanned by TapScanner

6.4 Hierarchical Agglomerative Clustering

- Hierarchical clustering does not require the specification of a pre-defined number of clusters. Fig. 6.4.1 shows clustering and hierarchical clustering.

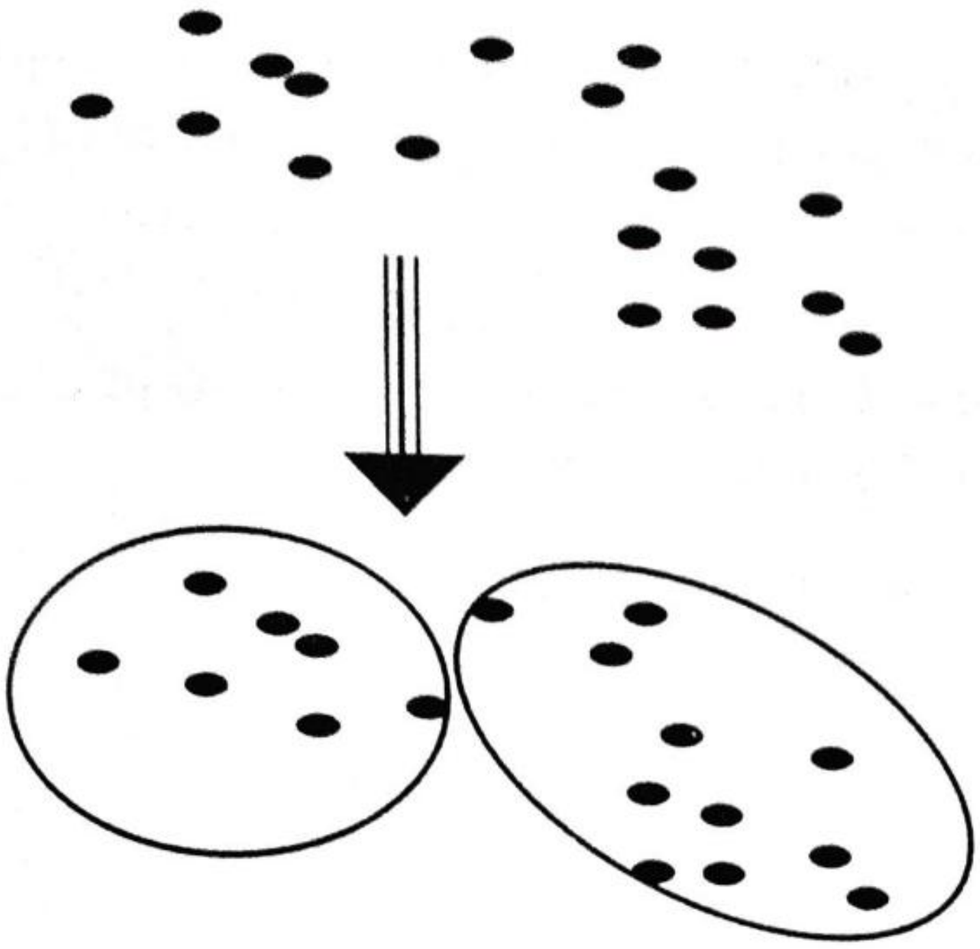


Fig. 6.4.1 (a) Clustering

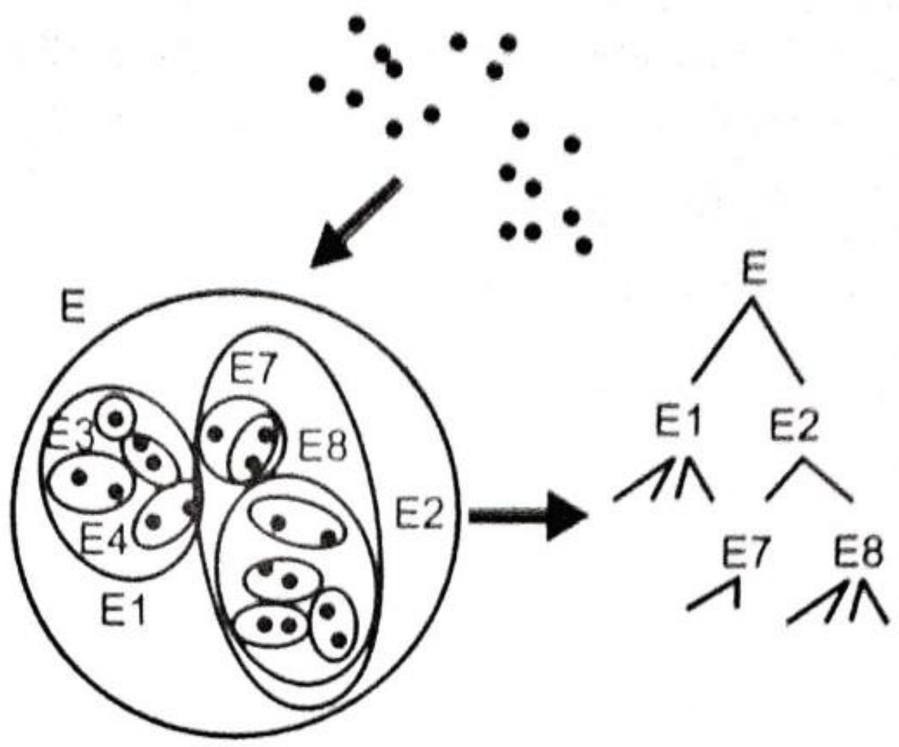


Fig. 6.4.1 (b) Hierarchical clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters k as an input, but needs a termination condition
- The core idea of hierarchical clustering is to create different levels of clusters. Each upper level cluster is conceptually the result of a merging of the clusters at lower levels. At the lowest level each cluster contains a single observation. At the highest level there is a single cluster that contains all observations.
- The construction of the levels is achieved by hierarchical clustering algorithms using a :
  1. Agglomerative (bottom-up) strategy
  2. Divisive (top-down) strategy

6.4.1 Agglomerative Clustering

- Agglomerative strategies start at the lowest level and recursively merge a selected pair of clusters into a single cluster of higher level. The pair chosen for merging is formed by the two groups with the smallest inter-group dissimilarity.
- It uses the Single-Link method and the dissimilarity matrix.

Steps :

1. Compute the Euclidean Distance between every pair of patterns. Let the smallest distance be between patterns X_i and X_j.

Scanned by TapScanner

2. Create a cluster C composed of X_i and X_j. Replace X_i and X_j by cluster vector C in the training set (the average of X_i and X_j).
3. Go back to 1, treating clusters as points, though with an appropriate weight, until no point is left.

- **Divisive (top-down) strategy :** Divisive methods recursively split one group into two new groups. The split is chosen that can achieve two groups with the highest inter-group dissimilarity. With both paradigms one obtains N-1 levels in the hierarchy.
- Introduced in Kaufmann and Rousseeuw. It is implemented in statistical analysis packages. Eventually each node forms a cluster on its own.

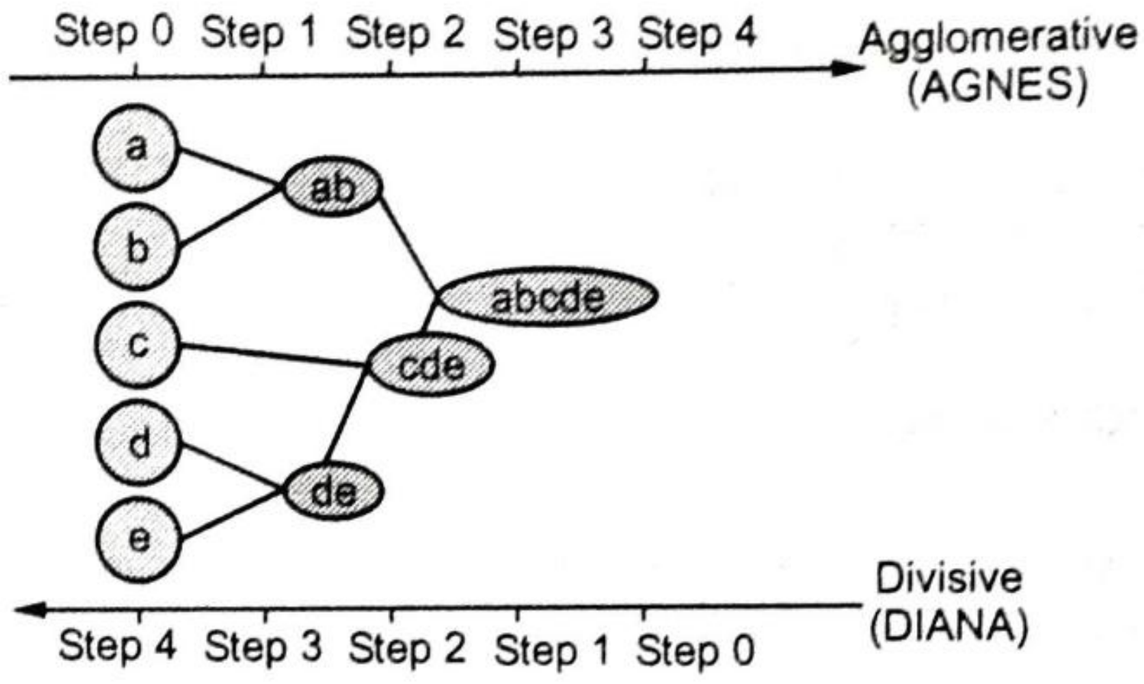


Fig. 6.4.2

- Each level of the hierarchy represents a specific grouping of the whole data set into disjoint clusters. The user must provide an additional criterion in order to extract from a hierarchical clustering an ordinary clustering : a criterion to pick the level of the hierarchy that corresponds to the natural clustering. One such criterion is the gap statistic.
- Since hierarchical clustering is a greedy search algorithm based on a local search, the merging decision made early in the agglomerative process are not necessarily the right ones. One possible solution to this problem is to refine a clustering produced by the agglomerative hierarchical algorithm to potentially correct the mistakes made early in the agglomerative process.

Method Based on Probabilities

1. Let us start with a tree whose root node contains all the training patterns and has a single empty successor.
2. Select a pattern X_i (if there are no more pattern to select, then terminate).
3. Set μ to the root node.

Scanned by TapScanner

4. For each of the successors of μ, calculate the best host for X_i, as a function of the **Z** value of the different potential partitions.
5. If the best host is an empty node η, then place X_i in η and generate an empty successor and sibling of η. Go to 2.
6. If the best host η is a non-empty singleton, then place X_i in η, generate successors $\{X_i\}$ and previous η to the new η, add empty successors everywhere, and go to 2.
7. If the best host is a non-empty, non-singleton node, η, place X_i in η, set μ to η, and go to 4.

- **Node Merging :** Attempt to merge the two best hosts. If merging improves the Z value, then do so.
- **Node Splitting :** Attempt to replace the best host among a group of sibling by that hosts's successors. If splitting improves the Z value, then do so.

6.4.2 Dendrogram

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram. A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.
- The agglomerative hierarchical clustering algorithms available in this program module build a cluster hierarchy that is commonly displayed as a tree diagram called a dendrogram.
- They begin with each object in a separate cluster. At each step, the two clusters that are most similar are joined into a single new cluster. Once fused, objects are never separated. The eight methods that are available represent eight methods of defining the similarity between clusters.

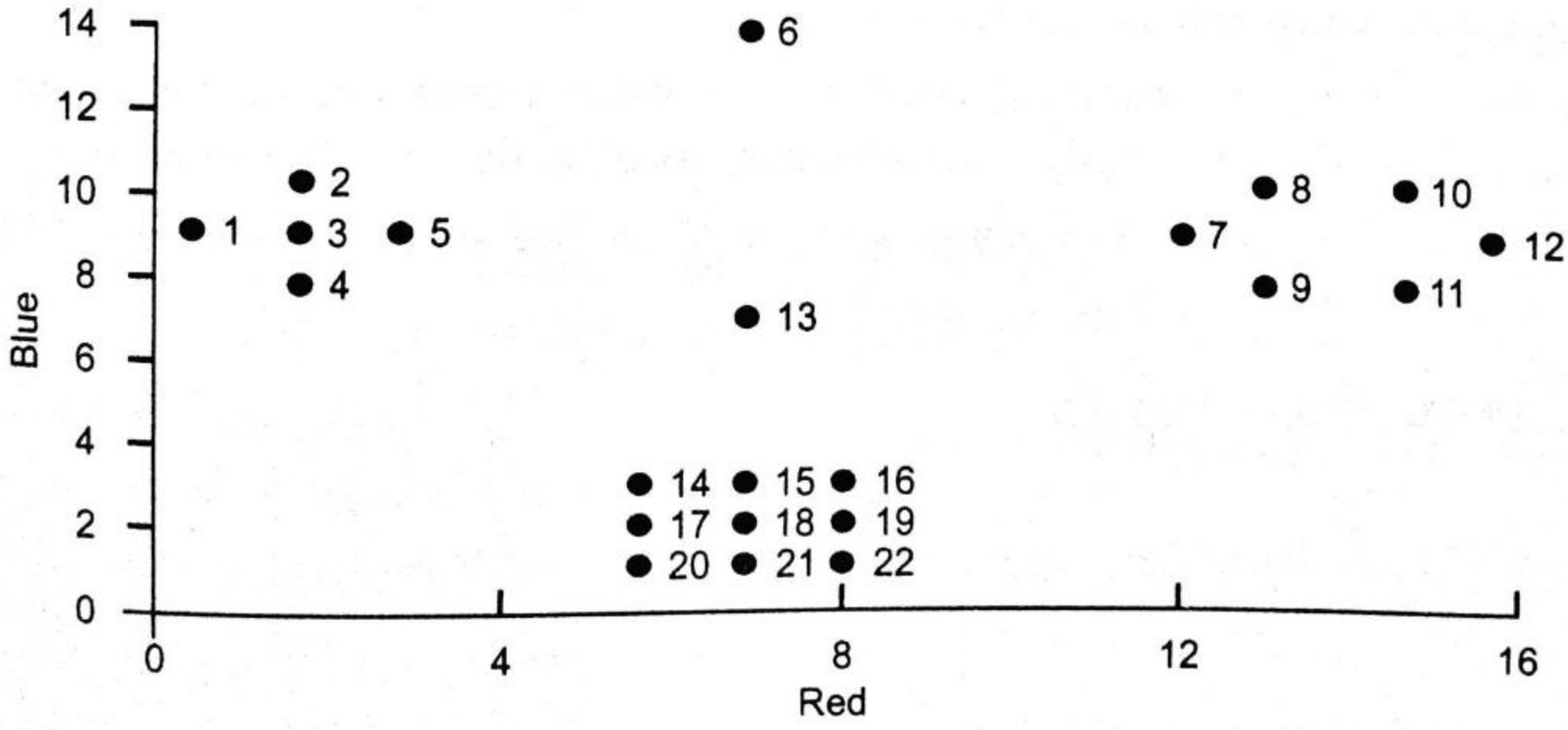


Fig. 6.4.3

Scanned by TapScanner

- Suppose we wish to cluster the bivariate data shown in the following scatter plot. In this case, the clustering may be done visually. The data have three clusters and two singletons, 6 and 13.
- The horizontal axis of the dendrogram represents the distance or dissimilarity between clusters. The vertical axis represents the objects and clusters. The dendrogram is fairly simple to interpret. Remember that our main interest is in similarity and clustering.
- Each joining of two clusters is represented on the graph by the splitting of a horizontal line into two horizontal lines. The horizontal position of the split, shown by the short vertical bar, gives the distance (dissimilarity) between the two clusters.

6.4.3 Difference between Hierarchical Clustering and K-Mean Clustering

Parameters	Hierarchical clustering	K-Mean clustering
Running Time	O(N3)	Fastest (each iteration is linear)
Assumptions	Requires a similarity / distance measure	Strong assumptions
Input parameters	None	K (number of clusters)
Clusters	Subjective (only a tree is returned)	Exactly K clusters

6.5 Clustering Terms using Documents

- The goal of a document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances. A distance measure thus lies at the heart of document clustering.
- The large variety of documents makes it almost impossible to create a general algorithm which can work best in case of all kinds of datasets.
- In the context of document clustering, a natural measure of similarity of two documents is the similarity between their word conditional distributions.
- Let X be the set of documents and let Y be the set of words, then for every document, we can define

$$p(y|x) = \frac{n(y|x)}{\sum_{y \in Y} n(y|x)}$$

where $n(y|x)$, is the number of occurrences of the word Y in the document x.

Scanned by TapScanner

Challenges in Document Clustering

1. Selecting appropriate features of the documents that should be used for clustering.
2. Selecting an appropriate similarity measure between documents.
3. Selecting an appropriate clustering method utilising the above similarity measure.
4. Implementing the clustering algorithm in an efficient way that makes it feasible in terms of required memory and CPU resources.
5. Finding ways of assessing the quality of the performed clustering.

6.6 Multiple Choice Questions

Q.1 Which of the following is hierarchical clustering method ?

a Agglomerative
b Divisive Clustering
c PAM
d A and B

Q.2 The k-means algorithm is sensitive to _____ because an object with an extremely large value may substantially distort the distribution of data.

a outliers
b text data
c boasting
d cluster

Q.3 _____ clustering method works by grouping data objects into a tree of clusters.

a PAM
b Density-based Method
c Hierarchical
d Grid-Based Method

Q.4 In DIANA, all of the objects are used to form _____ initial cluster.

a one
b two
c four
d eight

Q.5 if the clustering process is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called a _____.

a dendrogram
b nearest-neighbor clustering algorithm
c minimal spanning tree algorithm
d single-linkage algorithm

Scanned by TapScanner

Q.6 Which of the following is NOT type of clusters ?

a Well-separated clusters
b Prototype-based clusters
c Contiguity-based clusters
d DBSCAN clusters

Q.7 Shared Nearest Neighbors is a _____ clustering.

a density-based
b well-separated
c contiguity based
d graph based

Answer Keys for Multiple Choice Questions

Q.1	d	Q.2	a	Q.3	c	Q.4	a
Q.5	d	Q.6	d	Q.7	a		

❑❑❑

Scanned by TapScanner

7 Advanced Topics

Syllabus

Summarization, Topic detection and tracking, Personalization, Question answering, Cross language information retrieval.

Contents

7.1 Summarization

7.2 Topic Detection and Tracking

7.3 Personalization

7.4 Question Answering

7.5 Cross Language Information Retrieval

Scanned by TapScanner

7.1 Summarization

- Summarization systems generate condensed outputs that convey important information contained in one or more sources for particular users and tasks.
- Automatic summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content.
- In addition to text, images and videos can also be summarized. Text summarization finds the most informative sentences in a document; various methods of image summarization are the subject of ongoing research, with some looking to display the most representative images from a given collection or generating a video; video summarization extracts the most important frames from the video content.
- Multi-Document Summarization is a process of representing a set of documents with a short piece of text by capturing the relevant information and filtering out the redundant information.
- Two prominent approaches to Multi-Document Summarization are extractive and abstractive summarization. Extractive summarization systems aim to extract salient snippets, sentences or passages from documents, while abstractive summarization systems aim to concisely paraphrase the content of the documents.
- There are two methods to automatic summarization : Extraction and abstraction.
  1. Extraction-based summarization : Content is extracted from the original data, but the extracted content is not modified in any way. Examples of extracted content include key-phrases that can be used to "tag" or index a text document, or key sentences that collectively comprise an abstract, and representative images or video segments.
  2. Abstraction-based summarization : This has been applied mainly for text. Abstractive methods build an internal semantic representation of the original content, and then use this representation to create a summary that is closer to what a human might express. Abstraction may transform the extracted content by paraphrasing sections of the source document, to condense a text more strongly than extraction

7.2 Topic Detection and Tracking

- Topic Detection and Tracking (TDT) was a part of the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program. It was also closely related to the initiative known as The Text REtrieval Conference (TREC).

Scanned by TapScanner

- The data sets used in the TDT related contests and experiments are still available via the Linguistic Data Consortium (LDC).
- The problem of Topic Detection and Tracking consists in classifying incoming newswire and broadcast stories (documents) into groups concerning the same topics (categories). The stories are assumed to be coming from different sources and there may be multiple stories referring to the same topic/event at the same moment of time or evolving over some time period.
- The following basic concepts may be, however, distinguished :
  a) A story is a single document/newswire story conveying some information to the user; in general, a stream of stories on the input is assumed which are more or less explicitly separated one from another;
  b) An event is something what happens at a particular place and a particular time;
  c) A topic is an important event considered together with all related events
- These are namely the tracking of known events, the detection of unknown events, and the segmentation of a news source into stories.

1. **The segmentation task** : The segmentation task is defined to be the task of segmenting a continuous stream of text into its constituent stories. To support this task the story texts from the study corpus will be concatenated and used as input to a segmenter. This concatenated text stream will include only the actual story texts and will exclude external and internal tag information. The segmentation task is to correctly locate the boundaries between adjacent stories, for all stories in the corpus.
2. **The detection task** : The detection task is characterized by the lack of knowledge of the event to be detected. In such a case, one may wish to retrospectively process a corpus of stories to identify the events discussed therein, or one may wish to identify new events as they occur, based on an on-line stream of stories. Both of these alternatives are supported under the detection task.
3. **Retrospective event detection** : The retrospective detection task is defined to be the task of identifying all of the events in a corpus of stories.
4. **On-line new event detection** : The on-line new event detection task is defined to be the task of identifying new events in a stream of stories. Each story is processed in sequence, and a decision is made whether or not a new event is discussed in the story, after processing the story but before processing any subsequent stories.
5. **The tracking task** : The tracking task is defined to be the task of associating incoming stories with events known to the system.

Scanned by TapScanner

7.3 Personalization

- Personalized content retrieval aims at improving the retrieval process by taking into account the articular interests of individual users. It is well known that human preferences are complex, multiple, heterogeneous, changing, even contradictory, and should be understood in context with the user goals and tasks at hand
- Ambiguity means that a single ranking is unlikely to be optimal for all users. Personalized ranking is the only way to bridge the gap. Personalization can use
  a) Long term behavior to identify user interests, e.g., a long term interest in user interface research.
  b) Short term session to identify current task, e.g., checking on a series of stock tickers.
  c) User location, e.g., MTA in New York vs Baltimore.
  d) Social network.
- For each query q, compute average rating for each result. Let R_q be the optimal ranking according to the average rating. Then compute the NDCG value of ranking R_q for the ratings of each rater i.
- Let Avg_q be the average of the NDCG values for each rater. Let Avg be the average Avg_q over all queries. Potential for personalization is (1 – Avg).
- Personalization of retrieval is the approach that uses the user profiles, additionally to the query, in order to estimate the users' wishes and select the set of relevant documents.
- In this process, the query describes the user's current search, which is the local interest, while the user profile describes the user's preferences over a long period of time; we refer to the latter as global interest.
- The method for preference representation and extraction, as well as the estimation of the degree to which local or global interests should dominate in the selection of the set of relevant documents.
- Query operations techniques focus on the refinement of the need of information expressed by the user's input query. Implicit information like user preferences is used to reformulate the user queries.
- Users express their actual information need by a query. Query operations try to "modify" the given local interest so that the information need is somehow altered by the interests of the users.
- The idea of personalization through query operation is adding information from the user profile to bring the query closer, viewed as a vector model, to documents

Scanned by TapScanner

that are not only relevant to the query but also relevant to the preferences from the use.

- Link - based personalization is used because it follows the trend of link - based ranking algorithms. These have been a huge success in the past years, beginning with Google's PageRank. Link - based personalization affects directly to the document ranking techniques.
- Page rank values are often computed by web crawlers that start from an initial page, and do a random walking through the links of the page and the subsequent links of the pages pointed by the links. In general, link - based personalized algorithms are modifications of Google's PageRank or the HITS authority and hub algorithm.
- Fig. 7.3.1 shows Topology of Document weighting personalized search engines.

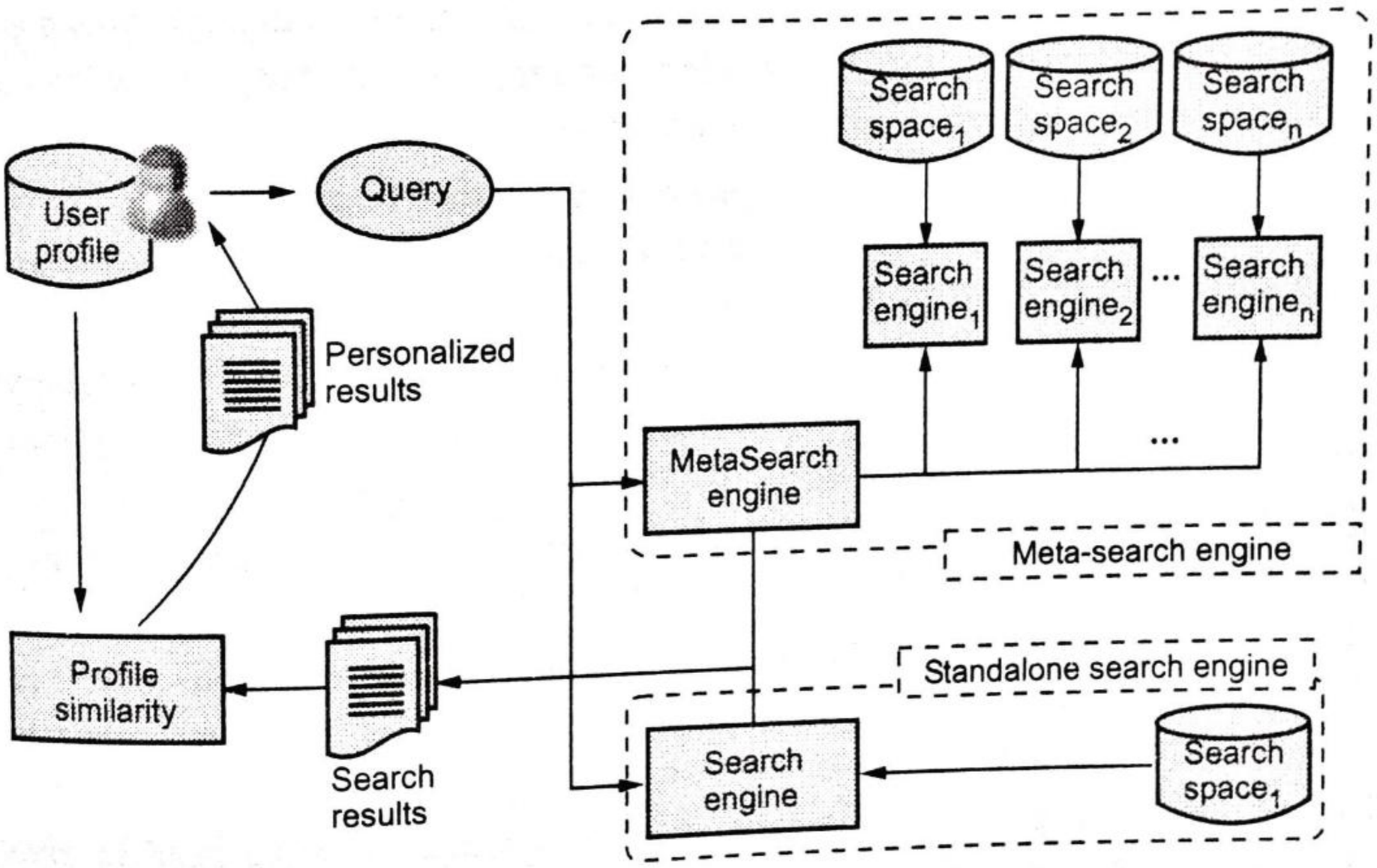


Fig. 7.3.1

- Document weighting algorithms modify the final ranking of the result set documents. Most search engines compute a ranking value of relevance between the document and the need of information.
- A personalized search engine can then compute a personalized ranking value for every document in the result set.
- The benefits of this approach is that this value has only to be computed for the returned top result set documents, the drawback is that this value has to be computed at query time.

Scanned by TapScanner

- This algorithm is also suitable for meta search engines, as commonly the user dependent algorithm can focus on a small quantity of the top returned documents being able to compute a personalize score at real - time, by accessing the document's content or even just using the provided summaries.
- The user - dependent score usually comes from a document - user similarity function, based on term - frequency similarity, classification and clustering algorithms, bayesian functions etc.

7.4 Question Answering

- Question answering is a sophisticated form of information retrieval characterised by information needs that are at least partially expressed as natural language statements or questions, and is one of the most natural forms of human computer interaction.
- In contrast with classical information retrieval, where complete documents are considered relevant to the information request, in question answering, specific pieces of information are returned as an answer.
- The user of a question answering system is interested in a concise, comprehensible and correct answer, which may refer to a word, sentence, paragraph, image, audio fragment, or an entire document.
- Question answering programs can construct answers through the query of a knowledge base (a structured database of knowledge) or an unstructured collection of documents in a natural language.
- Question answering systems are either closed domain or open domain. IBM's Watson is an example of the latter type of QA systems.
- Open domain answering systems take natural language questions and transform them into a structured query. Keyword extraction is used to determine the question type (who, where, how many).
- Part of speech tagging, and syntactic parsing techniques can be used to determine the answer type (person, location, number). Then an information retrieval system is used to find data containing the keywords. Parsing is then used to translate the answer into meaningful text.

7.5 Cross Language Information Retrieval

- Cross-Lingual Retrieval refers to the retrieval of documents that are in a language different from the one in which the query is expressed. This allows users to search document collections in multiple languages and retrieve relevant information in a form that is useful to them, even when they have little or no linguistic competence in the target languages.

Scanned by TapScanner

- Cross lingual information retrieval is important for countries like India where very large fraction of people are not conversant with English and thus don't have access to the vast store of information on the web.
- Fig. 7.5.1 shows cross lingual information retrieval. Example : A user may ask query in English but retrieve relevant documents written in Tamil.



Fig. 7.5.1 : Cross lingual information retrieval

- Accepting questions in one language (English) and retrieving information in a variety of other languages. Questions may be typical Web queries or full questions in across-lingual Question Answering (QA) system. Information could be news articles, text fragments or passages, factual answers, audio broadcasts, written documents, images, etc.
- Fig. 7.5.2 shows information retrieval cycle.

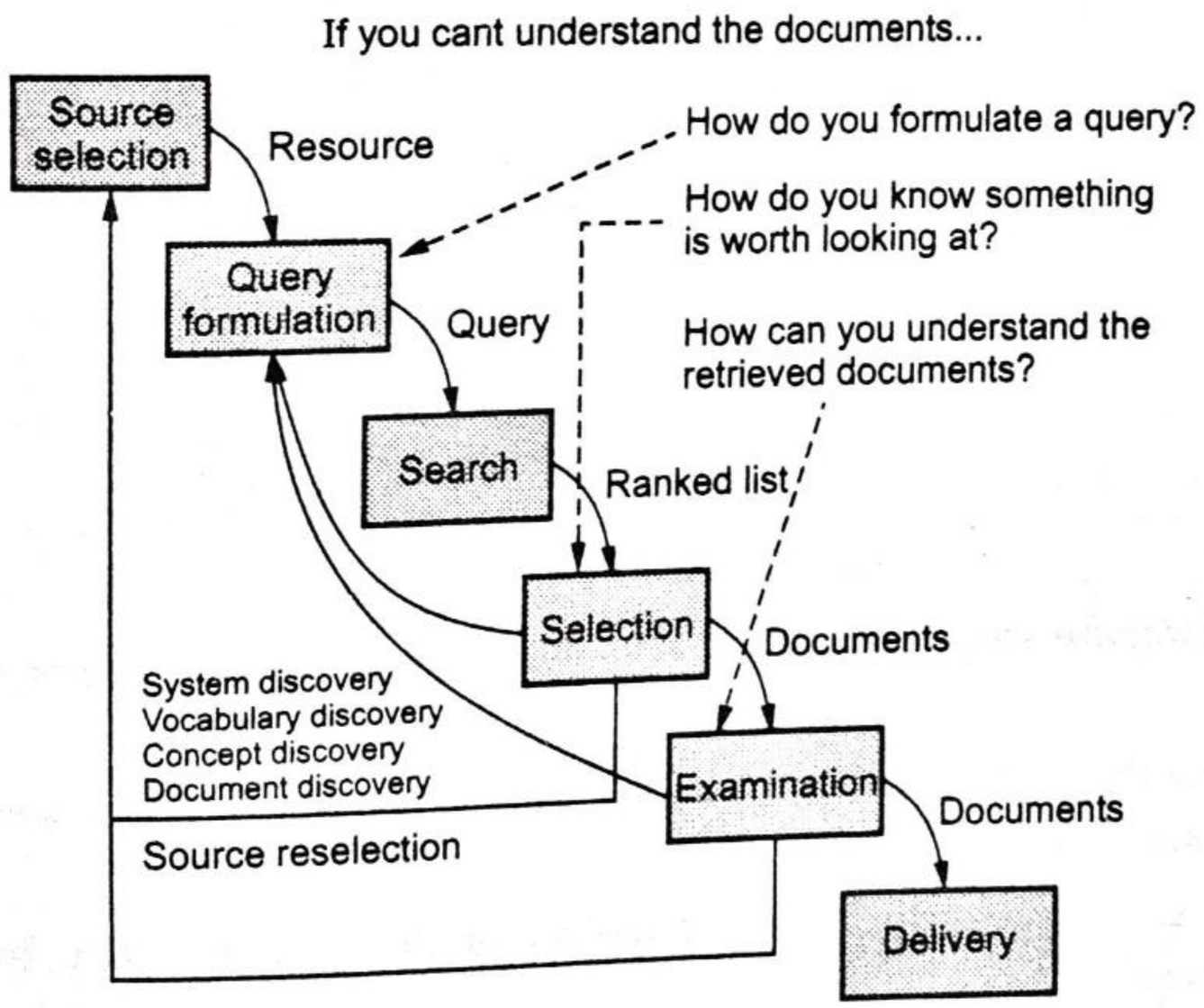


Fig. 7.5.2 Information retrieval cycle

- Importance of CLIR in digital libraries is extremely increasing in the Internet environment. It is more practical than translating every documents into every other languages.

Scanned by TapScanner

- A user can issue a query in the most fluent language and avoid multiple queries in each language. It would be useful for people who are not fluent enough to create a query in a foreign language, but who can read the language to understand document contents and judge their relevance.
- Two methods are used to solve this problem : query translation and document translation.

1. Query translation :

- Translate English query into french query
- Search Chinese document collection
- Translate retrieved results back into English

- Fig. 7.5.3 shows query translation.

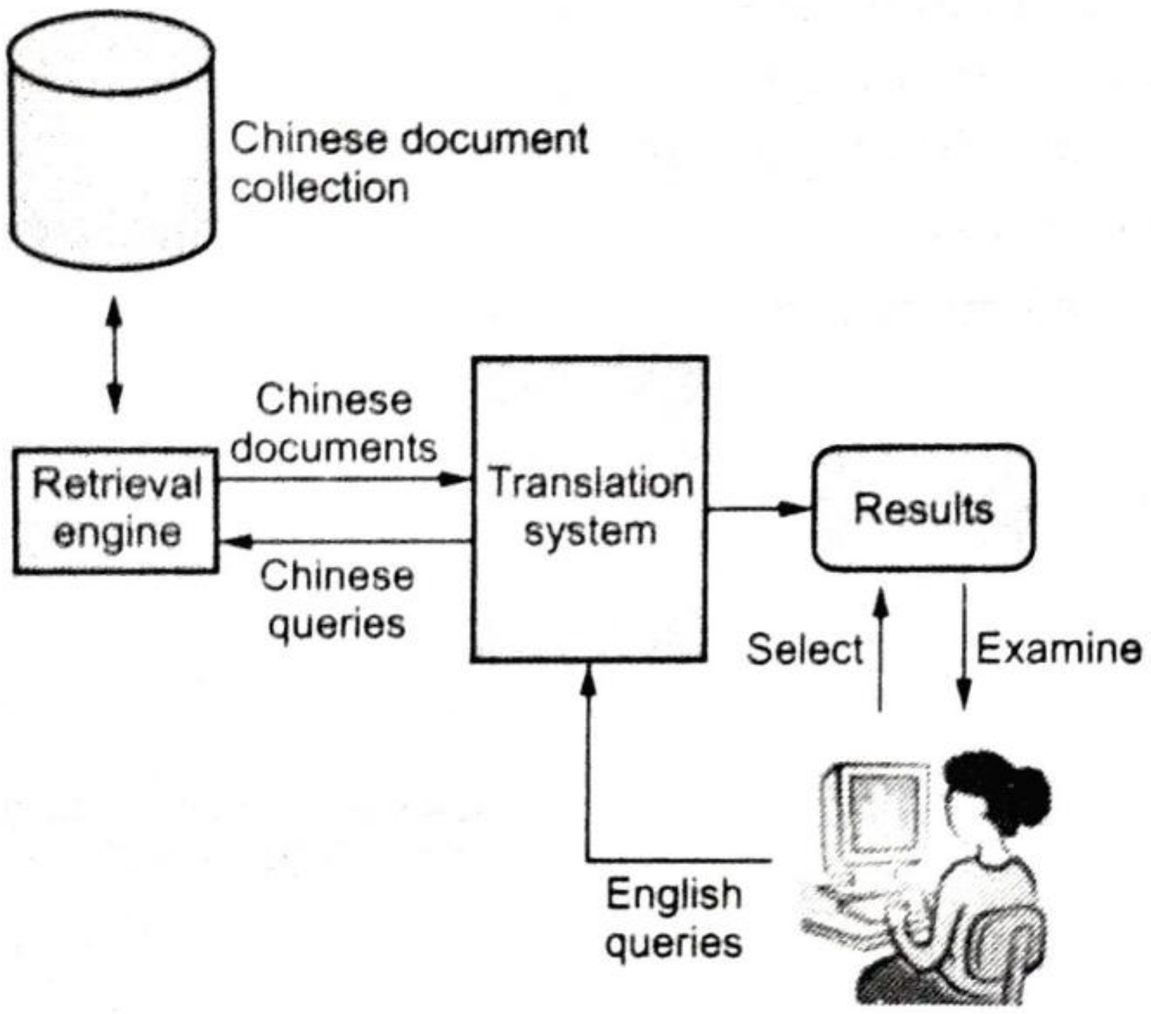


Fig. 7.5.3 Query translation

- Backend database stays unchanged and translation changes do not affect document base.
- Cross-linguality is transparent for the users. Fine-tuning between frontend and backend required.
- Query translation is easy. Disambiguation of query terms may be difficult with short queries.
- Translation of documents must be performed at query time.
- Pros of query translation :
  a. Flexibility (translation on demand)
  b. Less text to translate

Scanned by TapScanner

- Cons of query translation
  a. Less precise
  b. The retrieved documents need to be translated to be readable

2. Document translation :

- Translate entire document collection into English
- Search collection in English
- **Pros :**
  a) Translation may be (theoretically) more precise
  b) Documents become "readable" by the user
- **Cons :**
  a) Huge volume to be translated
  b) Impossible to translate them in all the languages
- **Challenges in creating a CLIR system :**
  1. Translation ambiguity : More than one translation may be possible while translating from source language to target language.
  2. Transliteration errors : Errors while transliteration might end up fetching the wrong word in target language.
  3. Dictionary coverage : For translations using bi-lingual dictionary, the exhaustiveness of the dictionary is important criteria for performance on system.
  4. Translate term : There are ambiguous names which need to be transliterated instead of translation.
  5. Morphological analysis is different for different languages.

❑❑❑

Scanned by TapScanner

8 Web Information Retrieval

Syllabus

Hypertext, web crawling, search engines, ranking, link analysis, PageRank, HITS.

Contents

8.1 *The Web*

8.2 *Hypertext*

8.3 *Web Crawling*

8.4 *Search Engines*

8.5 *Ranking*

8.6 *Link Analysis*

8.7 *PageRank & HITS*

8.8 *Multiple Choice Questions*

Scanned by TapScanner

8.1 The Web

- Web is collection of millions of files stored on thousands of servers all over the world. These files represent documents, pictures, videos, sounds, programs, interactive environments.
- Following are hardware, software, protocols that make up the web.
  1. A web server is a computer connected to the internet that runs a program that takes responsibility for storing, retrieving, distributing some of the web files. A web client (web browser) is a computer that requests files from the web.
  2. Well-defined set of languages and protocols independent of the hardware or operating system are required to run on the computers.
  3. The Hyper Text Markup Language (HTML) is the universal language of the Web.
  4. Java is a language for sending small applications over the web. Java script is a language for extending HTML to embed little programs called scripts in web pages. The primary purpose of Java and scripts is to speed up the interactivity of web pages.
  5. Pictures, drawings, charts, diagrams are displayed on web using image formats such as JPEG and GIF formats. The Virtual Reality Modeling Language is the web's way of describing three-dimensional objects.
- A web page is an HTML document that is stored on a web server. A Web site is a collection of web pages belonging to a particular organization. URL of these pages share a common prefix, which is the address of the size's home page.
- Search engines are a bottom-up approach for finding your way around the web. Some search engines search only the titles of web pages. While another search every word. Keywords can be combined with boolean operations, such as AND, OR , NOT, to produce rather complicated queries.

Web Browsers

- A web browser is a program. Web browser is used to communicate with web servers on the internet, which enables it to download and display the Web pages. Netscape Navigator and Microsoft Internet Explorer are the most popular browser software's available in the market.
- Browser interacts with web as well as computer operating system and with other programs. Most browser windows have the same basic layout. Some of the essential elements are menu bar, toolbar, address or location window, viewing window status bar.

Scanned by TapScanner

- The purpose of the web browser is to display web pages, which may either arrive over the internet. Web browser can be used to view files of any common web format that are stored on the user system.
- WWW uses client-server interaction. The browser program acts as a client that uses the internet to contact a remote server for a copy of the requested page. The server on the remote system returns a copy of the page along with the additional information.

8.1.1 Characteristics

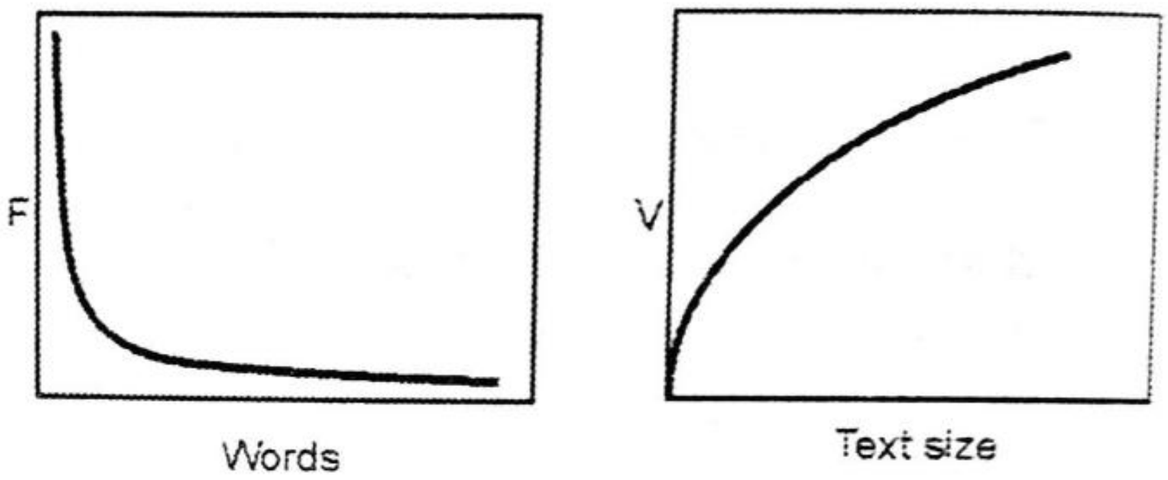
- In characterizing the structure and content of the web, it is necessary to establish precise semantics for web concepts.
- Two characteristics make retrieval of relevant information from the web a really hard task :
  a. The large and distributed volume of data available
  b. The fast pace of change
- Internet and particular web is dynamic in nature, so it is a difficult task to measure.
- In the mid-1980s, you might spend all afternoon visiting your friends before dropping by the bank and grocery, and then go out to dinner and a show after.
- Today, your banking, shopping and chat with your friends are all readily handled from your tablet or phone, and if you're not in the mood for a fancy outing, Netflix and a quick Internet-ordered pizza take care of the evening's entertainment needs and all of it can be accomplished in less time than it takes to say "The Internet made me a hermit."
- It was something of a privilege to be online, but the net gained traction really quickly. While just 16 million people were using the web in 1995, this figure had leapt to 50 million three years later, one billion by 2009 and more than twice that last year.
- In 1993 there was an estimated 130 websites online, which jumped to ten thousand by 1996. Fast-forward to 2012, and there are 634 million websites competing for your attention.

8.1.2 Modeling the Web

- Can we model the document characteristics of the whole web ? **Answer is "Yes".**
- The Heap's and Zipf's laws are also valid in the web. Normally the vocabulary grows faster and the word distribution should be more biased. But there are no such experiments on large Web collections to measure these parameters.

Scanned by TapScanner

- **Heaps' law :** An empirical rule which describes the vocabulary growth as a function of the text size.
- It establishes that a text of n words has a vocabulary of size $O(n^{\beta})$ for $0 < \beta < 1$.
- **Zipf's law :** An empirical rule that describes the frequency of the text words. It states that the i-th most frequent word appears as many times as the most frequent one divided by $i^{-\theta}$, for some $\theta > 1$.
- One more model is related to the distribution of document size. According to this model, the document sizes are self similar, they have a large variance. The probability of finding a document of size x bytes is given by :



(a) Distribution of sorted word frequencies

(b) Size of the vocabulary

Fig. 8.1.1

$$p(x) = \frac{1}{x_{\sigma}\sqrt{2\pi}} \exp - (ln\, x - \mu)^2\, 2\sigma^2$$

where, the average (μ) and standard deviation (σ) are 9.357 and 1.318 respectively.

- The majority of the documents are small, but there is a non trivial number of large documents. This is intuitive for image or video files, but it is also true for HTML pages. A good fit is obtained with the Pareto distribution

$$p(x) = \frac{\alpha\, k^{\alpha}}{x^{1+\alpha}}$$

where x is measured in bytes and k and α are parameters of the distribution.

- So what languages dominate the Web ? It should come as no surprise that english still dominates the web, with more than two-thirds of the web's pages being in english. According to a study by, a Web site in the language Catalan, Japanese is the second most popular language of Web sites.

Web pages by language

Language	Web pages	Percent of total
English	21,42,50,996	68.39
Japanese	1,83,35,739	5.85
German	1,80,69,744	5.77
Chinese	1,21,13,803	3.87

Scanned by TapScanner

French	92,62,663	2.96
Spanish	75,73,064	2.42
Russian	59,00,956	1.88
Italian	48,83,497	1.56
Portuguese	42,91,237	1.37
Korean	40,46,530	1.29
Dutch	31,61,844	1.01
Sweden	29,29,241	0.93

- As you can see, the growth is impressive and unimpeded. Also :
  a. The total number of web sites seems to follow Moore's Law and double every 18-24 months.
  b. At the current rate, we will hit 1 billion sites in 2013 and 2 billion sites in 2015.
  c. Over the years, the number of web sites seems to be roughly equal to the number of people on the internet.
  d. If WordPress continues on its current trajectory, there will be 300-500 million WordPress sites by 2015.

8.1.3 Link Analysis

- The goal of information retrieval is to find all documents relevant for a user query in a collection of documents. With the advent of the web new sources of information became available, one of them being the hyperlinks between documents and records of user behaviour.
- Collections of documents connected by hyperlinks. Hyperlinks provide a valuable source of information for web information retrieval. This area of information retrieval is commonly called link analysis.
- A hyperlink is a reference of a web page B that is contained in web page A. When the hyperlink is clicked on in a web browser, the browser displays page B. This functionality alone is not helpful for web information retrieval.
- Link analysis has been used successfully for deciding which web pages to add to the collection of documents and how to order the documents matching a user query.
- Link analysis for web search has intellectual antecedents in the field of citation analysis, aspects of which overlap with an area known as bibliometrics.

Scanned by TapScanner

- Link analysis are of three types : Microscopic level, mesoscopic level and macroscopic level.
  1. Microscopic level : It is related to the statistical properties of links and individual nodes.
  2. Mesoscopic level : It is related to the properties of areas or regions of the web.
  3. Macroscopic level : It is related to the structure of the web at large.

8.2 Hypertext

- Hypertext systems allow for non-sequential, or non-linear, reading. This is the underlying idea of a hypertext system. The result is a multidimensional document that can be read by following different paths through it.
- The main use of hypertext is in information retrieval applications. The ease of linking different pieces (fragments) of information is the important aspect of hypertext information retrieval. The information can be of various media: it may be fragments of textual documents, structured data from databases, or list of terms and their definitions. Any of these, or a mixture thereof, can make up the contents of a hypertext document. Therefore, in a hypertext system it is possible to :
  a) link with a term that represents aspects of the content of a document
  b) connect two related documents
  c) relate a term to a fragment containing its definition and use
  d) link two related terms
- A hypertext document contains links referring to other parts of the document, or even to whole other documents. A hypertext document does not have to be read serially; the fragments of information can be accessed directly via the links contained in the document.
- The links embedded in a document are known as hyperlinks. When selected, these hyperlinks allow for the portion of the document linked to by the hyperlink to be displayed. This allows the reader to jump to another part of the same page, another page in the same document, or another document. By following a series of hyperlinks, the reader can follow their own path through the document.

8.3 Web Crawling

- A web crawler is a program which browses the World Wide Web in a methodical, automated manner. Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine that will index the downloaded pages to provide fast searches.
- Web crawlers is also called ant, bot, worm or web spider. The process of scanning the WWW is called Web crawling or spidering.

Scanned by TapScanner

- Web crawling is the process by which we gather pages from the web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them.
- It starts with a list of URLs to visit. As it visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, recursively browsing the Web according to a set of policies.
- Web crawlers are mainly used to create a copy of all the visited pages for later processing by a search engine, which will index the downloaded pages to provide fast searches.
- Crawlers can also be used for automating maintenance tasks on a web site, such as checking links or validating HTML code. Also, crawlers can be used to gather specific types of information from web pages, such as harvesting e-mail addresses.
- When a search engine's web crawler visits a web page, it "reads" the visible text, the hyperlinks, and the content of the various tags used in the site, such as keyword rich meta tags.
- Using the information gathered from the crawler, a search engine will then determine what the site is about and index the information. The website is then included in the search engine's database and its page ranking process.

Crawling process :

- The crawler begins with a seed set of URLs to fetch. The crawler fetches and parses the corresponding web-pages and extracts both text and links. The text is fed to a text indexer; the links (URL) are added to a URL frontier.
- Initially, the URL frontier contains the seed set; as pages are fetched, the corresponding URLs are deleted from the URL frontier. The entire process may be viewed as traversing the web graph. In continuous crawling, the URL of a fetched page is added back to the frontier for fetching again in the future.

8.3.1 Web Crawler Architecture

- Fig. 8.3.1 shows web crawler architecture.
- The **URL frontier,** containing URLs yet to be fetched in the current crawl. In the case of continuous crawling, a URL may have been fetched previously but is back in the frontier for re-fetching.
- A **DNS resolution** module that determines the web server from which to fetch the page specified by a URL.
- A fetch module that uses the http protocol to retrieve the web page at a URL.
- A parsing module that extracts the text and set of links from a fetched web page.

Scanned by TapScanner

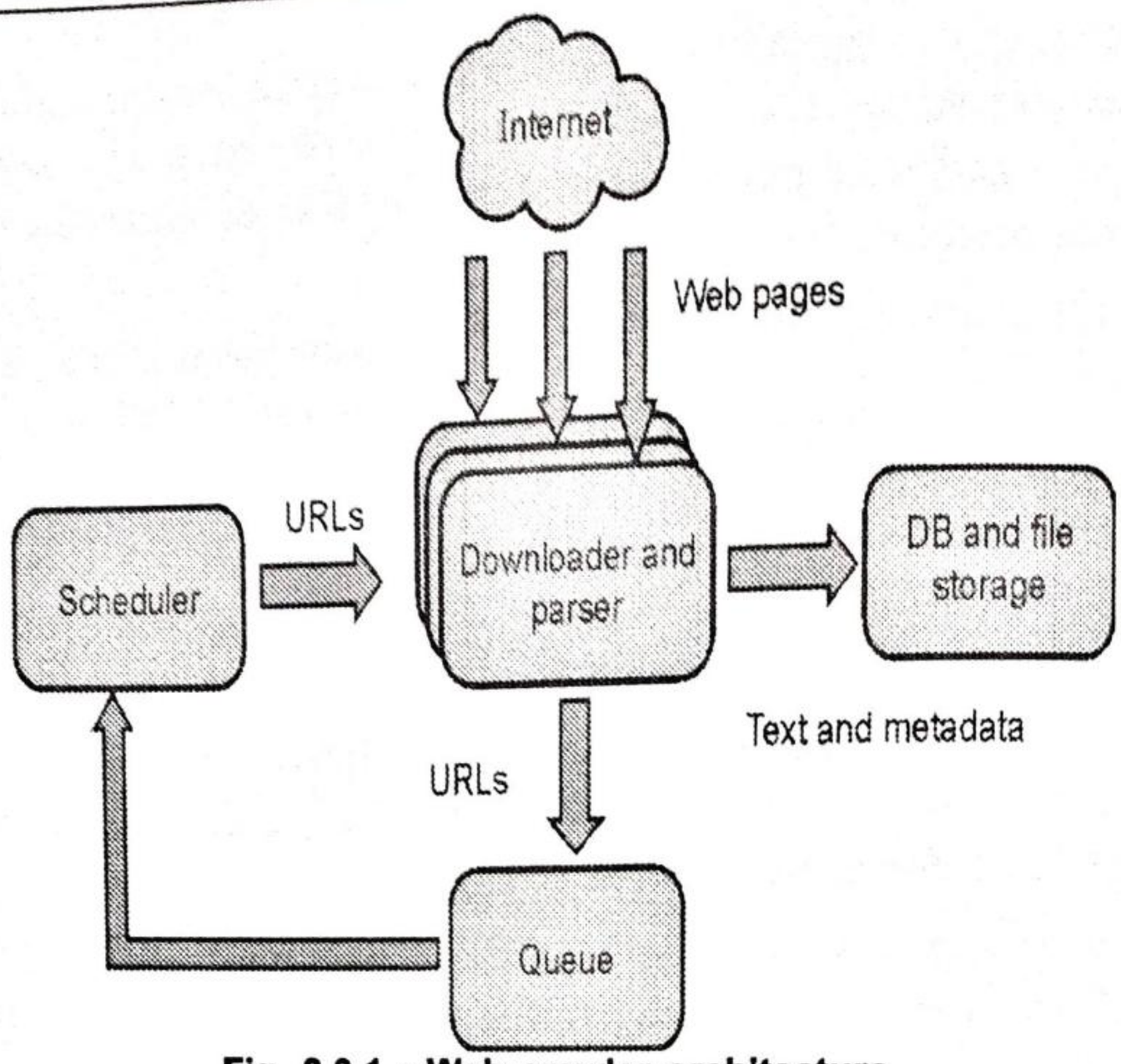


Fig. 8.3.1 : Web crawler architecture

- A duplicate elimination module that determines whether an extracted link is already in the URL frontier or has recently been fetched.
- **Robots.txt file :** A web server administrator can use a file / robots.txt to notify a crawler what is allowed and what is not.

Politeness policies :

- Within website www.cs.sfu.ca, many pages contain many links to other pages in the same site. Downloading many pages simultaneously from one website may overwhelm the server.
- A reasonable web crawler should use only a tiny portion of the bandwidth of a website server, not fetching more than one page at a time.
- **Implementation :** Logically split the request queue into a single queue per web server; a server queue is open only if it has not been accessed within the specified politeness window.
- Suppose a web crawler can fetch 100 pages per second and the politeness policy dictates that it cannot fetch more than 1 page every 30 seconds from a server then we need URLs from at least 3,000 different servers to make the crawler reach its peak throughput.
- **Detecting updates :** If a web page is updated, the page should be crawled again. In HTTP, request HEAD returns only header information about the page but not the page itself. A crawler can compare the date it received from the last GET request with the Last-Modified value from a HEAD request.

Scanned by TapScanner

8.3.2 Taxonomy of Crawler

Features of a crawler :

1. Robustness : Ability to handle spider-traps (cycles, dynamic web pages). Robustness should not fall into spider traps which generate an infinite number of pages.
2. Politeness : Policies about the frequency of robot visits.
3. Distribution : Crawling should be distributed within several machines.
4. Scalability : Crawling should be extensible by adding machines, extending bandwidth, etc.
5. Efficiency : Clever use of the processor, memory, bandwidth.
6. Quality : Should detect the most useful pages, to be indexed first.
7. Freshness : Should continuously crawl the web.
8. Extensibility : Should support new data formats (e.g. XML-based formats), new protocols (e.g. ftp), etc.

Web page Types :

- **Static web pages** don't change content or layout with every request to the web server. They change only when a web author manually updates them with a text editor or web editing tool like Adobe Dreamweaver.
- Dynamic web pages can adapt their content or appearance depending on the user's interactions, changes in data supplied by an application, or as an evolution over time, as on a news web site.

8.4 Search Engines

- Search engines are becoming the primary entry point for discovering web pages. Ranking of web pages influences which pages users will view. Exclusion of a site from search engines will cut off the site from its intended audience.
- A search engine is a program designed to help find information stored on a computer system such as the World Wide Web or a personal computer. The search engine allows one to ask for content meeting specific criteria and retrieves a list of references that match those criteria.

8.4.1 Cluster based Architecture

- Centralized crawler indexer architecture is used by most of the search engine. Using crawlers, information is gathered into a single site, where it is indexed; the site then processes all user queries.

Scanned by TapScanner

- This system has its own data collecting mechanism, and all the data are stored and indexed in a conventional database system. Although many web search engines download web pages and provide service by thousands of servers, they all belong to this kind according to their basic architecture.
- Centralized architecture consists of following components :
  1. Crawlers 2. Index 3. Query engine 4. User interface.

1. Crawlers

- Crawlers are programs i.e. software agents that traverse the web sending new or updated pages to a main server where they are indexed. It sends requests to web sites and downloads Web pages. The local system sends requests for web pages to remote Web servers.
- In the downloaded documents it discovers new pages and new servers. The effect of such repeated requests is of a "robot" that moves from site to site, gathering information at every site it visits. In reality, this "crawler" never leaves the local system.
- Crawlers are also called robots, spiders, wanderers and walkers.
- Fig. 8.4.1 shows the software architecture of a search engine based on Altavista architecture.

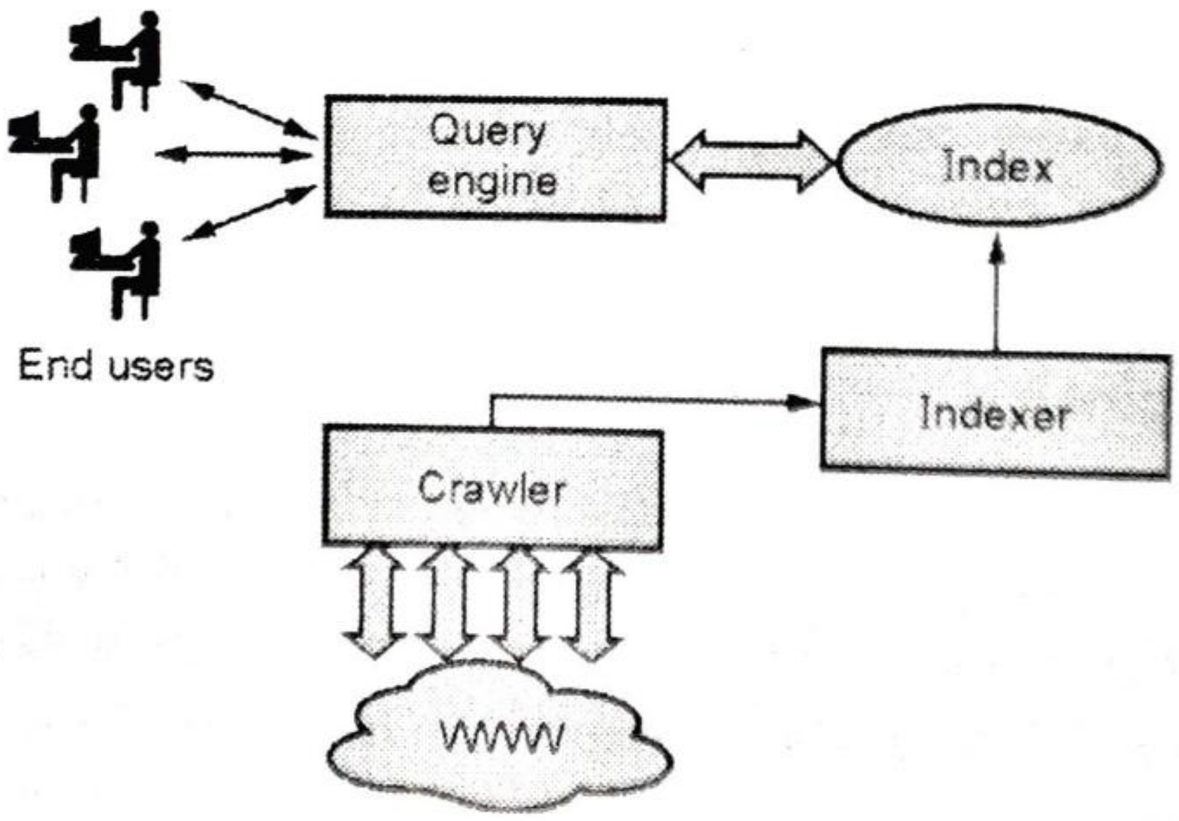


Fig. 8.4.1 : Typical crawler indexer architecture

- Figure consists of two parts : One deal with users, consisting of the user interface and the query engine and another that consists of the crawler and indexer modules.

2. Indexer

- The index is used in a centralized fashion to answer queries submitted from different places in the web. Index the downloaded pages. Each downloaded page is processed locally.

Scanned by TapScanner

- The indexing information is saved and the page is discarded.
- A module that takes a collection of documents or data and builds a searchable index from them. Common practices are inverted files, vector spaces, suffix structures and hybrids of these.
- *Exception* : Some search engines keep a local cache copy of many popular pages indexed in their database, to allow for faster access and in case the destination server is temporarily in-accessible.

3. User interface :

Solicit queries and deliver answers. All requests are submitted to a single site.

4. Query engine :

- It process queries against the index. All processing is done locally. It requires a massive array of computers and storage.
- Following table gives idea about search engine with URL and web page indexed upto May 1998.

Search engine	URL	Web page indexed
AltaVista	www.altavista.com	140
AOL Netfind	www.aol.com/netfind/	-
Excite	www.excite.com	55
Google	google.stanford.edu	25
GoTo	goto.com	-
HotBot	www.hotbot.com	110
Infoseek	www.infoseek.com	30
Lycos	www.lycos.com	30
Magellan	www.mckinley.com	55
Microsoft	search.msn.com	-
NorthernLight	www.nlsearch.com	67
WebCrawler	www.webcrawler.com	2

Problems using this architecture :

1. Difficult to gathering the data because of dynamic nature of the web.
2. The high load on web servers.
3. Large volume of data : Could be difficult for crawler to cope with web growth
4. Communication link problem.

Scanned by TapScanner

- Most of the search engines are based in the United States and focuses on document in English. Some search engines specialized in different countries and different languages. Some of the search engines retrieve only specific web pages such as personal or institutional home pages or specific objects such as electronic mail address, images etc.

8.4.2 Distributed Architecture

- When the data source is large enough that even the metadata can't be efficiently managed in a database system, we can choose distributed system. Distributed information retrieval system does not have its own actual record database. It just indexes the interface of sub database system.
- When receiving a query from a user, main system will instantly obtain the records from sub databases through their search interfaces. The limitation of this system is that the number of sub databases can't be many, otherwise the search speed can't be ensured. A famous system is InfoBus system in Stanford digital library project.
- Harvest is an example of distributed architecture.

Harvest

- Harvest is a distributed crawler-indexer architecture which addresses the main problems in crawling and indexing the web :
  1. Web servers get requests from different crawlers of search engines which increase the server's load;
  2. Most of the entire objects retrieved by the crawlers are useless and discarded;
  3. No coordination exists among the crawlers.
- Harvest is designed to be a distributed search system where machines work together to handle the load which a single machine could not handle.
- Harvest also can be used to save bandwidth by deploying gatherers near the data source and exchanging the summarized data which usually is much smaller than the original data.
- But it seems most of further harvest applications are in the field of caching web objects instead of providing advanced internet search services. State of the art indexing techniques can reduce the size of an inverted file to about 30 % of the size of the text.
- Fig. 8.4.2 shows Harvest architecture.

Scanned by TapScanner

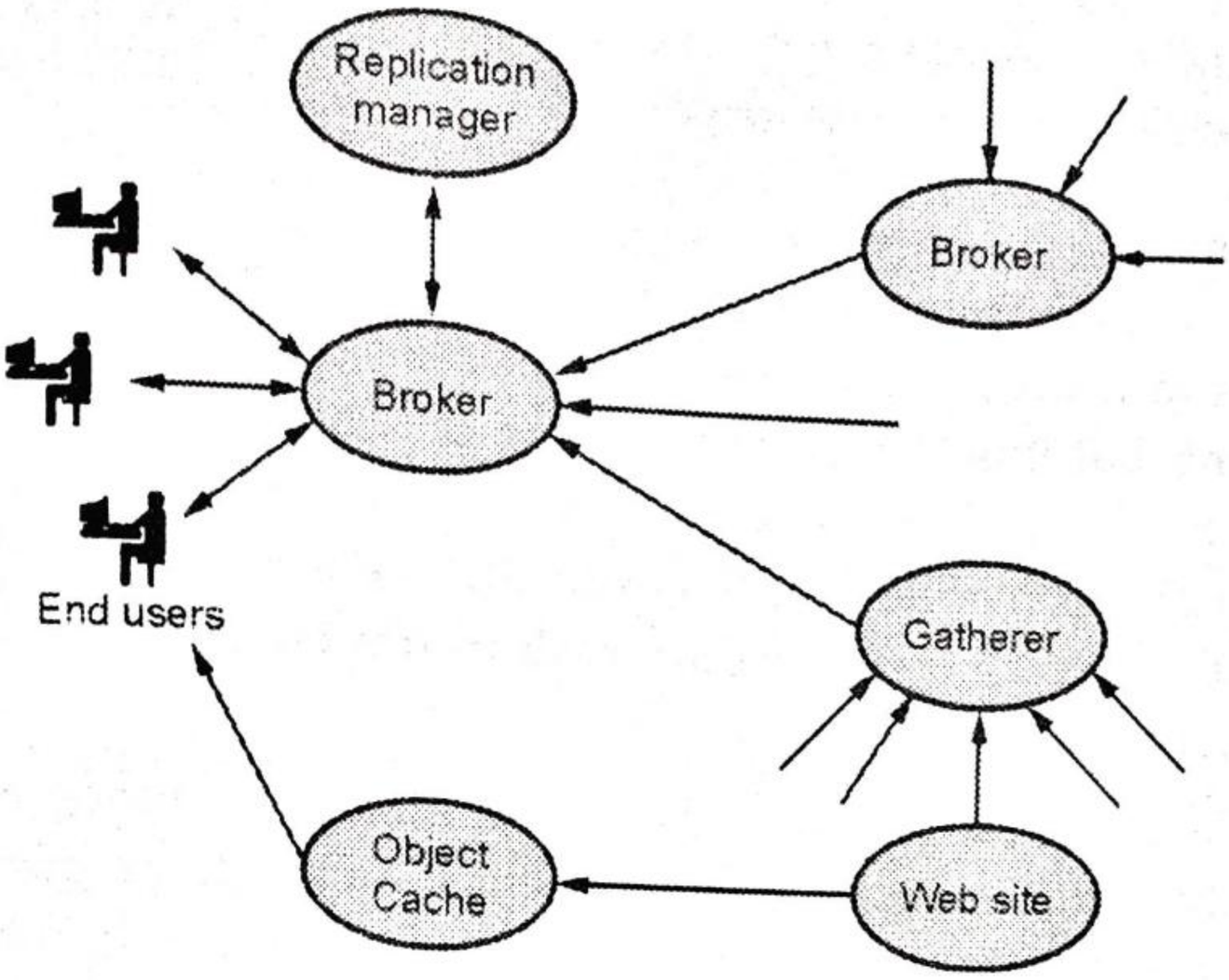


Fig. 8.4.2 : Harvest architecture

- Harvest does not suffer from some of common problems of the crawler-indexer architectures, such as
  1. Increased servers load caused by reception of simultaneous requests from different crawlers
  2. Increased web traffic, due to crawlers retrieving entire objects, while most content is not retained eventually
  3. Lack of coordination between engines, as information is gathered independently by each crawler

Goals of harvest :

- One of the goals of the harvest is to build topic specific brokers, focusing the index contents and avoiding many of the vocabulary and scaling problems of generic indices. It includes a distinguished broker that allows other brokers to register information about gathers and brokers.
- This is useful to search for an appropriate broker or gatherer when building a new system.

Components :

1. Gatherers : It extracts information from the documents stored on one or more web servers. It can handle documents in many formats like HTML, PDF, Postscript, etc.
2. Broker : Provides the indexing mechanism and query interface.
3. Replicator : To replicate servers. It enhances user base scalability.

Scanned by TapScanner

4. Object cache : Reduces network and server load. It also reduces response latency when accessing web pages.

Features of harvest :

- Harvest is a modular, distributed search system framework with working set components to make it a complete search system. The default setup is to be a web search engine, but it is also much more and provides following features :
  1. Harvest is designed to work as distributed system. It can distribute the load among different machines. It is possible to use a number of machines to gather data. The full-text indexer doesn't have to run on the same machine as broker or web server.
  2. Harvest is designed to be modular. Every single step during collecting data and answering search requests are implemented as single programs. This makes it easy to modify or replace parts of Harvest to customize its behaviour.
  3. Harvest allows complete control over the content of data in the search database. It is possible to customize the summarizer to create desired summaries which will be used for searching. The filtering mechanism of Harvest allows making modifications to the summary created by summarizers. Manually created summaries can be inserted to the search database.
  4. The Search interface is written in perl to make customization easy, if desired.
- For 100 million pages, this implies about 150 GB of disk space. Assuming that 500 bytes are required to store the URL and the description of each web page, we need 50 GB to store the description for 100 million pages.
- The use of meta search engines is justified by coverage studies that show that a small percentage of web pages are in all search engines. Moreover, less than 1 % of the web pages indexed by AltaVista, HotBot, Excite and Inforseek are in all of those search engines.

Advantages of distributed architecture :

1. *Server load reduced :* A gatherer running on a server reduces the external traffic (i.e., crawler requests) on that server.
2. *Network traffic reduced :* Crawlers retrieve entire documents, whereas Harvest moves only summaries.
3. *Redundant work avoided :* A gatherer sending information to multiple brokers reduces work repetition.

Scanned by TapScanner

8.5 Ranking

Ranking refers to the process search engines use to determine where a particular piece of content should appear on a SERP. Search visibility refers to how prominently a piece of content is displayed in search engine results.

8.5.1 Link based Ranking

- The primary challenge of a search engine is to return results that match a user's needs. A word will potentially map to millions of documents. How to order them ?
- Most of the search engines use variations of the boolean or vector model to do ranking. Every search engine strives to return relevant web pages that will satisfy your requests. Each search engine uses a proprietary 'ranking algorithm' that attempts to instantly build a list of highly appropriate responses to your query.
- Since each search engine applies its own formula to a unique database of information, results and relevancy rankings will always vary from search engine to search engine.
- Yuwono and Lee propose three ranking algorithms in addition to the classical scheme. They are called
  1. Boolean spread 2. Vector spread 3. Most cited
- Boolean spread and vector spread are the normal ranking algorithms of the boolean and vector model extended to include pages pointed to by a page in the answer. The last most cited is based only on the terms included in pages having a link to the pages in the answer.
- Hyperlink information is also used by some of the new ranking algorithms. The number of hyperlinks that point to a page provides measures of its popularity and quality. Also many links in common between pages or pages referenced by the same page often indicates a relationship between those pages. Here we will discuss three examples of ranking techniques.

a. WebQuery :

- It allows visual browsing of web pages. WebQuery takes a set of web pages and ranks them based on how connected each web page is.

b. HITS algorithm :

- Algorithm developed by Kleinberg in 1998.
- Second method is used in Hypertext Induced Topic Search (HITS). This ranking scheme depends on the query and considers the set of pages S that point to or are pointed by pages in the answer.

Scanned by TapScanner

- Pages that have many links pointing to them in S are called authorities. Pages that have many outgoing links are called hubs. Gives each page a hub score and an authority score. A good authority is pointed to by many good hubs. A good hub points to many good authorities. Users want good authorities.
- Computes hubs and authorities for a particular topic specified by a normal query. First determines a set of relevant pages for the query called the base set S. Analyze the link structure of the web sub-graph defined by S to find authority and hub pages in this set.
- Hubs : It contains many outward links and lists of resources.
- Authorities : It contains many inward links and provides resources, content.

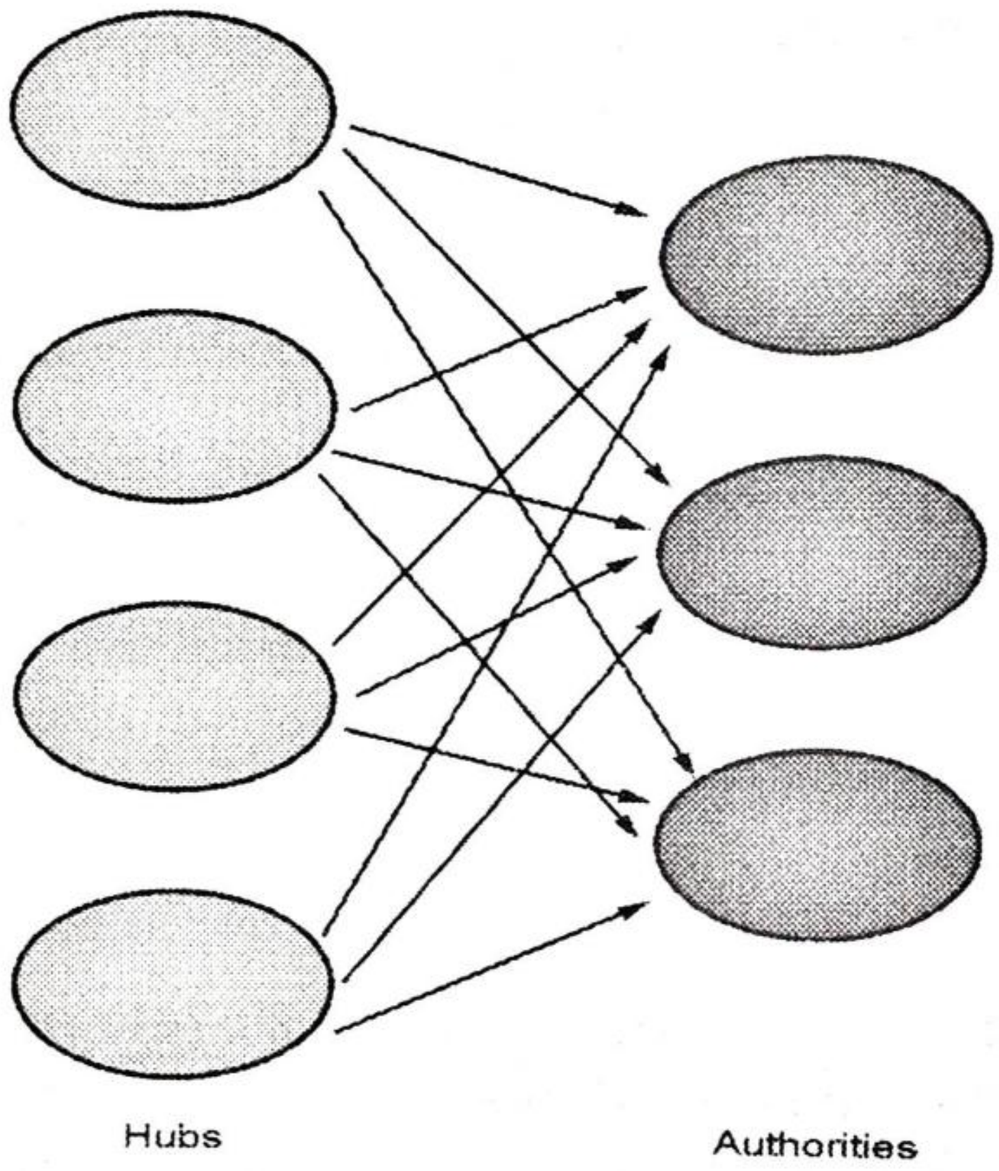


Fig. 8.5.1

- Let H(p) and A(p) be the hub and authority value of pages p. These values are defined such that the following equations are satisfied for all pages p. Authorities are pointed to by lots of good hubs :

$$a_p = \sum_{q : q \to p} h_q$$

- Hubs point to lots of good authorities :

$$h_p = \sum_{q : p \to q} a_q$$

C. PageRank

- Numeric value to measure how important a page is. PageRank (PR) is the actual ranking of a page, as determined by Google.

Scanned by TapScanner

- A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50 % chance" of something happening. Hence, a PageRank of 0.5 means there is a 50 % chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.
- Purpose : To increase the quality of search results and has left all of its competitors for dead.
- The Google Page Rank is based on how many links you have to and from your pages, and their respective page rank.
- We update our index every four weeks. Each time we update our database of web pages, our index invariably shifts : We find new sites, we lose some sites, and sites ranking may change. Your rank naturally will be affected by changes in the ranking of other sites.
- The Google PageRank (PR) is calculated for every webpage that exists in Google's database. It's real value varies from 0,15 to infinite, but for representation purposes it is converted to a value between 0 and 10 (from low to high). The calculation of the PR for a page is based on the quantity and quality of web pages that contain links to that page.
- Let C(a) be the number of outgoing links of page "a" and suppose that page "a" is pointed to by pages p1 to pn. Then, the PageRank PR(a) of "a" is defined as

$$PR(a) = 1 + (1 - q) \sum_{i=1}^{n} PR(p_i)/C(p_i)$$

Where q must be set by the system.

- It is obvious that the PageRank algorithm does not rank the whole website, but it's determined for each page individually. Furthermore, the PageRank of page A is recursively defined by the PageRank of those pages which link to page A.
- PageRank can be computed using an iterative algorithm. This means that each page is assigned an initial starting value and the PageRanks of all pages are then calculated in several computation circles based on the equations determined by the PageRank algorithm. The iterative calculation shall again be illustrated by the three-page example, whereby each page is assigned a starting PageRank value of 1.

8.5.2 Simple Ranking Functions

- Simplest ranking scheme use a global ranking function such as PageRank. In this case, quality of a web page in the result set is independent of the query. The query only selects pages to be ranked.
- To elaborate ranking scheme, use a linear combination of different ranking signals.

Scanned by TapScanner

- To illustrate, consider the pages p that satisfy query Q Rank score R (p, Q) of page p with regard to query Q can be computed as :

$$R(p, Q) = \alpha\ BM25(p, Q) + (1 - \alpha)\ PR\ (p)$$

Where $\alpha = 1$: text-based ranking, early search engines

$\alpha = 0$: link-based ranking, independent of the query

- Current engines combine a text-based ranking with a link-based ranking.

8.5.3 Learning to Rank

- Various methods are used for computing a web ranking. Some of them are given below :
  1. Apply machine learning techniques to learn the ranking of the results.
  2. Use a learning algorithm fed with training data that contains ranking information.
  3. Loss function to minimize : number of mistakes done by learned algorithm.
- Given query Q, three types of training data can be used :
  1. Pointwise : A set of relevant pages for Q
  2. Pairwise : A set of pairs of relevant pages indicating the ranking relation between the two pages
  3. Listwise : A set of ordered relevant pages : $p_1 > p_2 \cdots > p_m$

8.5.4 Evaluations

- To evaluate quality, web search engines typically use,
  a. Human judgements of which results are relevant for a given query.
  b. Some approximation of a ground truth inferred from user's clicks.
  c. Combination of both.
- To evaluate search results, use precision-recall metrics. Precision of web results should be measured only at the top positions in the ranking, say P@5, P@10, and P@20. Each query-result pair should be subjected to 3-5 independent relevant assessments.
- An advantage of using click-through data to evaluate the quality of answers derives from its scalability. Note that users' clicks are not used as a binary signal but in significantly more complex ways such as :
  a. Considering whether the user remained a long time on the page it clicked (a good signal),

Scanned by TapScanner

b. Jumped from one result to the other (a signal that nothing satisfying was found), or

c. The user clicked and came back right away (possibly implies web spam)

- These measures and their usage are complex and kept secret by leading search engines. An important problem when using clicks is to take into account that the click rate is biased by the ranking of the answer and the user interface.

8.6 Link Analysis

- The goal of information retrieval is to find all documents relevant for a user query in a collection of documents. With the advent of the web new sources of information became available, one of them being the hyperlinks between documents and records of user behavior. Collections of documents connected by hyperlinks. Hyperlinks provide a valuable source of information for web information retrieval. This area of information retrieval is commonly called **link analysis.**
- A hyperlink is a reference of a web page B that is contained in web page A. When the hyperlink is clicked on in a web browser, the browser displays page B. This functionality alone is not helpful for web information retrieval.
- Link analysis has been used successfully for deciding which web pages to add to the collection of documents and how to order the documents matching a user query.
- Link analysis for web search has intellectual antecedents in the field of citation analysis, aspects of which overlap with an area known as bibliometrics.
- The use of link analysis techniques to estimate the importance of a page made web search engines useful, and was central to the meteoric rise of Google as the leading commercial engine.
- In **query-independent ranking** a score is assigned to each page without a specific user query with the goal of measuring the intrinsic quality of a page. At query time this score is used with or without some query-dependent criteria to rank all documents matching the query.
- In **query-dependent ranking** a score measuring the quality and the relevance of a page to a given user query is assigned to some of the pages.
- Link analysis can also be used for deciding which web pages to add to the collection of web pages, i.e., which pages to crawl. A crawler performs a traversal of the web graph with the goal of fetching high-quality pages. After fetching a page, it needs to decide which page out of the set of uncrawled pages to fetch next. One approach is to crawl the pages with highest number of links from the crawled pages first.

Scanned by TapScanner

- Link analysis was also used for a search-by-example approach to searching : given one relevant page find pages related to it.

8.6.1 Finding Needle in the Haystack

- Haystack, an object storage system optimized for the efficient storage and retrieval of billions of photos.
- Typically, users expect a search query to complete in a short time, which makes it impossible to evaluate a complex ranking model on each document in the corpus, and so a two-phase scheme is used. First, a small number of potentially relevant documents are identified using simpler retrieval models which permit fast query evaluation. In the second phase, a more accurate but computationally expensive machine-learned model is used to re-rank these documents.
- Training data is used by a learning algorithm to produce a ranking model which computes relevance of responses for actual queries.
- Three different search strategies
  1. The correct answer is the title of the document that answers the question.
  2. The title of a document that answers the question is in the question itself.
  3. The answer-justifying document is a third-party document whose title is neither the answer nor in the question.
- Searching and browsing are the two major paradigms for information discovery. Users generally search for content when they are looking for something in specific, while they browse when they are interested in seeing what is available and discovering new content.
- In the past, the two paradigms were mostly thought of as separate and users had to choose between searching or browsing. Recently however, there has been a shift towards integrating these two ways of finding information, by making search results easier to browse and adding search functionality to browsing interfaces.
- Search activities can be broken down into three major categories : lookup, learning and investigating.
- Many search engines have recently started making their search results more personalized by integrating information from users' social network. For example, Google now displays social results, which are mixed in with regular results and are annotated with their relationship to the user.
- The Haystack architecture consists of 3 core components : The Haystack Store, Haystack Directory and Haystack Cache. Fig. 8.6.1 shows Haystack architecture for facebook.

Scanned by TapScanner

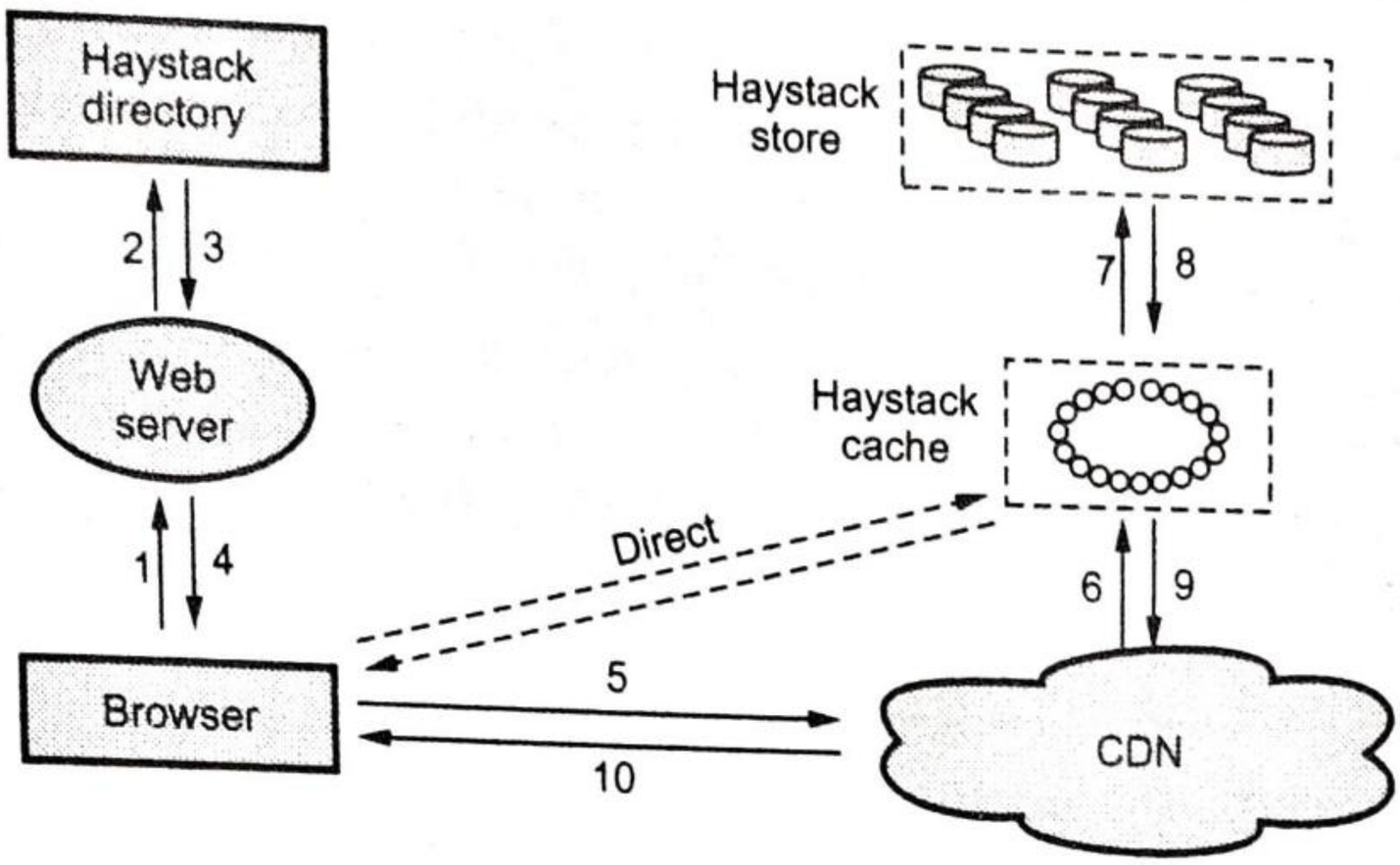


Fig. 8.6.1

- Directory is a relatively straight-forward component that stores its information in a replicated database accessed via a PHP interface.
- Haystack Cache : The Cache receives HTTP requests for photos from CDNs and also directly from users' browsers.
- Each Store machine manages multiple physical volumes. Each volume holds millions of photos.
- Haystack provides a fault-tolerant and simple solution to photo storage at dramatically less cost and higher throughput than a traditional approach using NAS appliances.
- Haystack is incrementally scalable.

8.6.2 Hubs and Authorities

- Given a user query, the algorithm first iteratively computes a hub score and an authority score for each node in the neighborhood graph. The documents are then ranked by hub and authority scores, respectively.
- *Authorities* are pages that are recognized as providing significant, trustworthy, and useful information on a topic. In-degree (number of pointers to a page) is one simple measure of authority. However in-degree treats all links as equal.
- *Hubs* are index pages that provide lots of useful links to relevant content pages (topic authorities). Hub pages for IR are included in the home page.
- Documents that have high authority scores are expected to have relevant content, whereas documents with high hub scores are expected to contain hyperlinks to relevant content.

Scanned by TapScanner

- A document which points to many others might be a good hub, and a document that many documents point to might be a good authority. Recursively, a document that points to many good authorities might be an even better hub, and similarly a document pointed to by many good hubs might be an even better authority.
- Authorities are pages containing useful information like course home pages and home pages of car manufacturers. Hubs are pages that link to authorities like course bulletin and list of india car manufacturers. Fig. 8.6.2 shows hub and authorities.

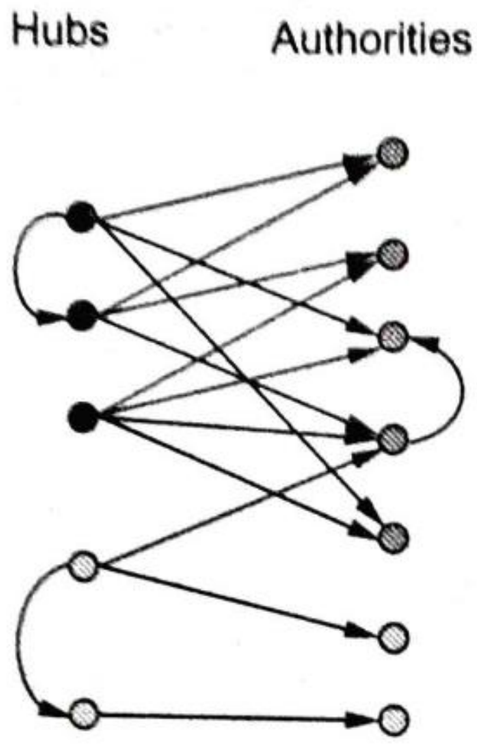


Fig. 8.6.2 Hub and authority

- Fig. 8.6.3 shows calculating Hub and Authority Value.

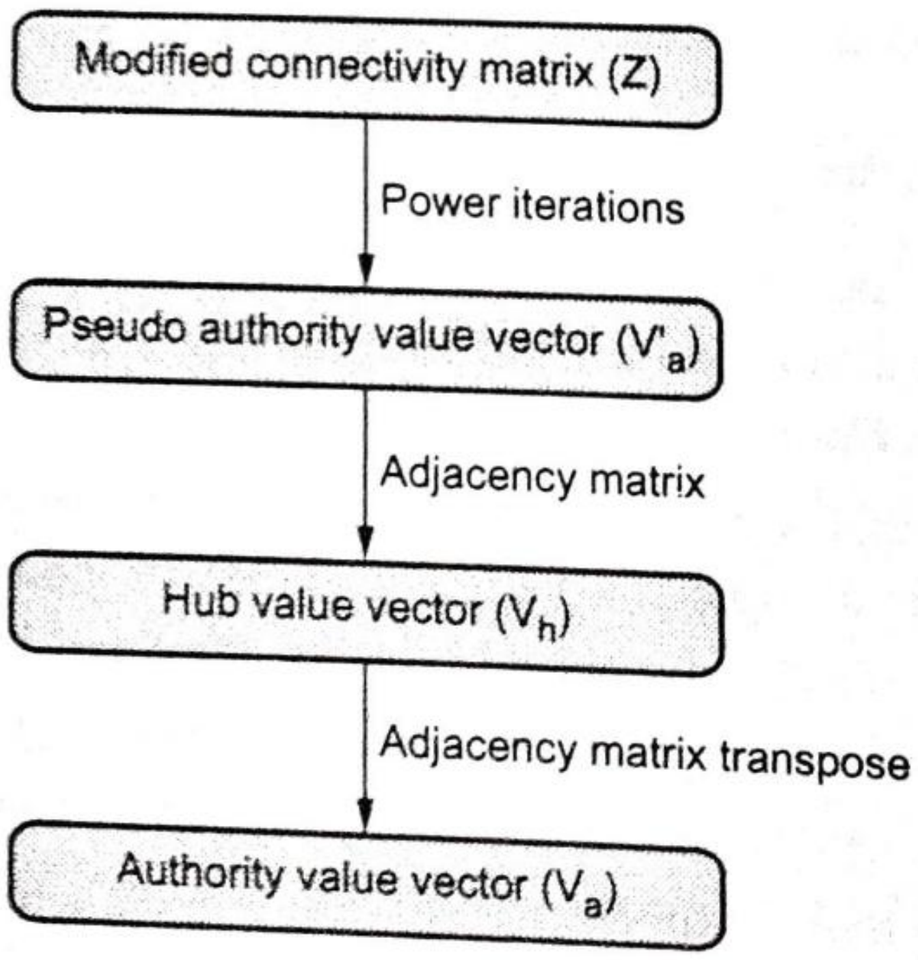


Fig. 8.6.3 Hub and authority

Scanned by TapScanner

1. Authority Update Rule : For each page p, update auth(p) to be the sum of the hub scores of all pages that point to it
2. Hub Update Rule : For each page p, update hub(p) to be the sum of the authority scores of all pages that it points to.

8.7 PageRank & HITS

- Two popular algorithms were introduced in 1998 to rank web pages by popularity and provide better search results. They are:
  1. HITS (Hypertext Induced Topic Search)
  2. Page Rank

8.7.1 Page Rank Algorithm

- PageRank was developed by Larry Page and Sergey Brin. It is first as part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.
- It is a link analysis algorithm. Page rank algorithm is non-keyword specific.
- PageRank is a numeric value that represents the importance of a page present on the web. When one page links to another page, it is effectively casting a vote for the other page.
- The Google search engine uses the PageRank algorithm as well. It simulates users using the search engine and applies the equation to rank the Web pages.
- The rank value indicates an importance of a particular page. A hyperlink to a page counts as a vote of support.
- A web page is important if it is pointed to by other important web pages. Google calculates a page's importance from the votes cast for it.
- Importance of each vote is taken into account when a page's PageRank is calculated.
- PageRank is Google's way of deciding a page's importance. It matters because it is one of the factors that determines a page's ranking in the search results.
- PageRank(PR) is the probability of a page being visited by such user under this model. For each web page, Page Rank value is pre-computed.
- PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size.
- A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening.

Scanned by TapScanner

- Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.
- Types of Links :
  1. **Inbound links or Inlinks :** Inbound links are links into the site from the outside. Inlinks are one way to increase a site's total Page Rank. Sites are not penalized for inlinks.
  2. **Outbound links or Outlinks :** Outbound links are links from a page to other pages in a site or other sites.
  3. **Dangling links :** Dangling links are simply links that point to any page with no outgoing links.
- Here a page obtains a higher rank if sum of its back-links is high. This algorithm is based on random surfer model.
- The random surfer model assumes that a user randomly keeps on clicking the links on a page and if she/he get bored of a page then switches to another page randomly.

Random Surfer Model

- A random surfer is one who bounces along randomly following the hyperlink structure of the Web. The random surfer chooses one of the several outlinks present in a page.
- The importance of the page is determined by the proportion of time spent by the surfer on that page. Brin and Page used this notion of random surfer to describe the adjustments made to their basic model.
- A surfer browsing the web by randomly following links, occasionally jumping to a random page. Fig. 8.7.1 shows random surfer model.

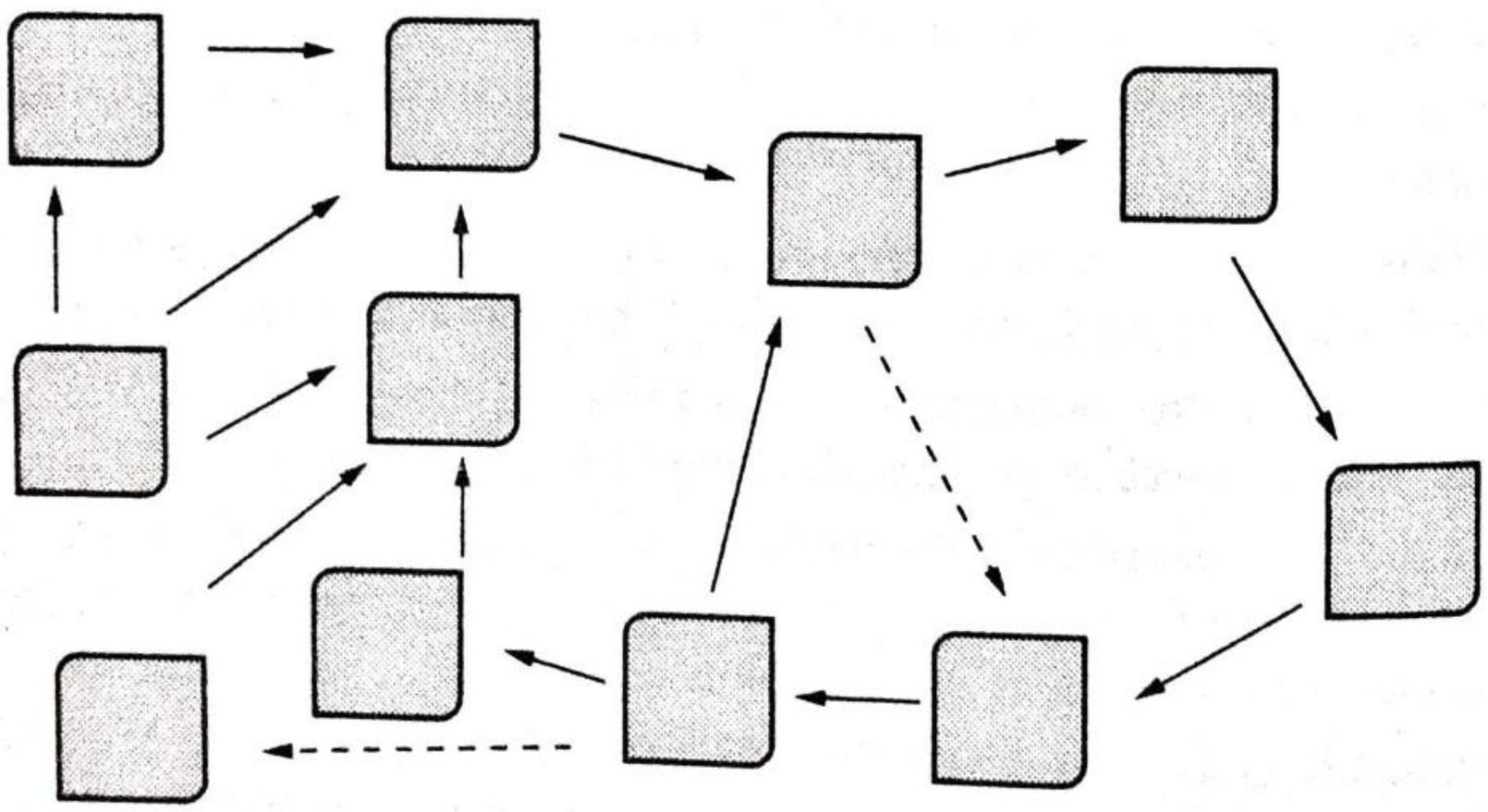


Fig. 8.7.1 : Random surfer model

Scanned by TapScanner

- The random surfer visits a web page with a certain probability which derives from the page's PageRank.
- The probability that the random surfer clicks on one link is solely given by the number of links on that page.
- This is why one page's PageRank is not completely passed on to a page it links to, but is devided by the number of links on the page.
- So, the probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page.
- This probability is reduced by the damping factor d. The justification within the random surfer model, therefore, is that the surfer does not click on an infinite number of links, but gets bored sometimes and jumps to another page at random.
- There are two problems with the random surfer model :
  1. A random surfer is caught when he encounters a dangling node such as an image, pdf, data tables etc.
  2. A random surfer completely abandons the hyperlink method and moves to a new browser and enters the URL in the URL line of the browser.
  - The modified model : The "random surfer" simply keeps clicking successive links at random, but periodically "gets bored" and jumps to a random page based on the distribution of E.

The damping factor (d) :

- The probability for the random surfer not stopping to click on links is given by the damping factor d, which depends on probability therefore, is set between 0 and 1.
- The higher d is, the more likely will the random surfer keep clicking links. Since the surfer jumps to another page at random after he stopped clicking links, the probability therefore is implemented as a constant (1-d) into the algorithm.
- Regardless of inbound links, the probability for the random surfer jumping to a page is always (1-d), so a page has always a minimum PageRank.
  - Some of the additional factors which can influence PageRank are :
    1. Visibility of a link
    2. Position of a link within a document
    3. Importance of a linking page
    4. Up-to-date of a linking page

Scanned by TapScanner

- The page rank algorithm divides the page rank value equally to all the outlinking web pages but all outlinks are not equally important or relevant to user query. This problem can be solved by considering an additional weight factor in the calculation of page rank and gives birth to a new algorithm known as weighted page rank algorithm.
- Searching with PageRank : Two search engines :
  a. Title-based search engine
  b. Full text search engine

Title-based search engine

- It searches only the "Titles". Finds all the web pages whose titles contain all the query words.
- Sorts the results by PageRank.
- Very simple and cheap to implement.
- Title match ensures high precision, and PageRank ensures high quality.

Full text search engine

- Also called Google. It examines all the words in every stored document and also performs PageRank.
- More precise but more complicated.

Advantages of PageRank

1. PageRank is a global measure and is query independent.
2. Less susceptibility to localized links.

Disadvantages of PageRank

1. It is calculated and stored at the time of indexing and not at the time of query.
2. The Rank sinks problem occurs when in a network pages get in infinite link cycles.

8.7.2 HITS

- HITS stands for Hypertext Induced Topic Search. HITS is search query dependent.
- HITS is a link analysis algorithm developed by Jon Kleinberg in 1998 to rate Web pages. The HITS algorithm is an iterative algorithm developed to quantify each page's value as an authority and as a hub.
- The most popular social networking website twitter makes use of HITS algorithm in suggesting user accounts to follow.
- HITS algorithm is executed at the query time rather than at the time of indexing

Scanned by TapScanner

- When the user issues a search query, HITS first expands the list of relevant pages returned by a search engine and then produces two rankings of the expanded set of pages, **authority ranking** and **hub ranking**.
- It categorizes a web page in two ways :
- Authority : pages that provide important and trustworthy information on a given topic. So an authority is a page that is pointed by many hubs.
- Hub : pages that contain links to authorities i.e pointing to many pages.
- HITS algorithm associates two scores with each page. Authority which is a measure of authoritative information contained in the page and hub which is a measure of links to good authorities.
- Using some existing search system, HITS algorithm gets a set of relevant pages for user query. This is referred to as the root set. Then all the pages from one link neighborhood of the root set are added to the root set. This is referred to as the base set.
- A good hub points to many good authorities, and a good authority is pointed to by many good hubs. Authorities and hubs have a mutual reinforcement relationship.
- HITS algorithm is having following two steps :
  1. Sampling step : collect the set of relevant pages for the given query.
  2. Iterative step : Found the hubs and authorities using output of step 1
- Fig. 8.7.2 shows HITS algorithm.

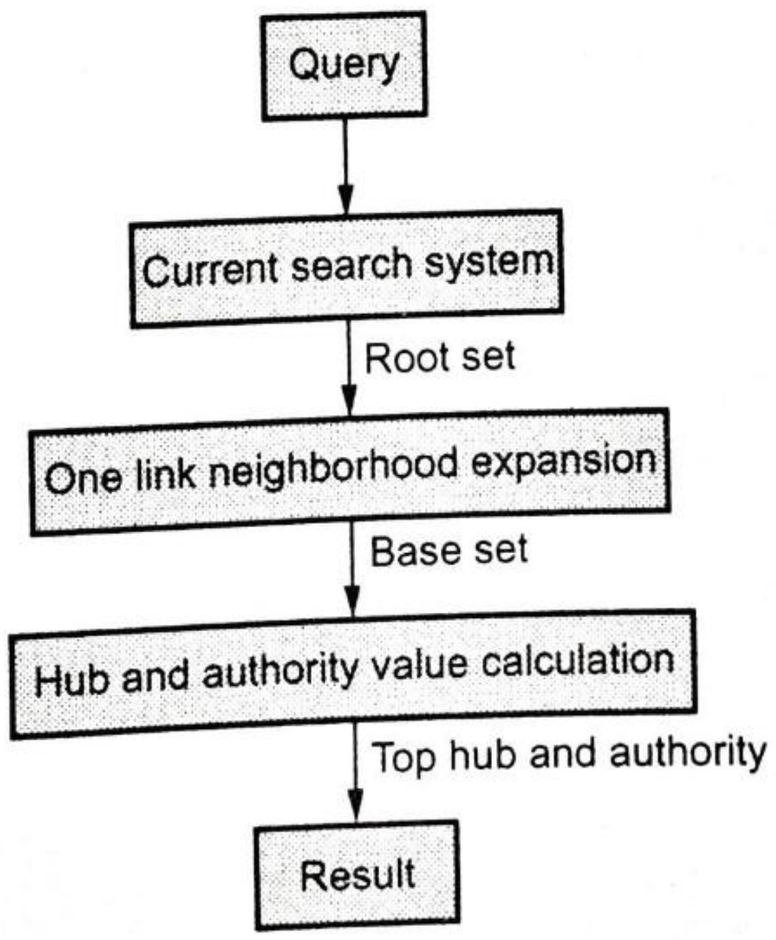


Fig. 8.7.2

Scanned by TapScanner

- After calculating hub and authority values on the root set, we select top 20 hubs and top 20 authorities for expansion. Good hubs point to good authorities, so we add out links of top 20 hubs to the root set. Good authorities are pointed by good hubs, so we add inlinks of top 20 authorities to the root set.
- As compared to HITS algorithm, this selective expansion procedure drastically reduces size of the base set and avoids topic drift as irrelevant pages are not added to the root set.
- A web page in HITS tends to have a hub weight and authority weight. An authority page is one that contains a lot of information about the topic. A hub page is one that contains a large number of links to pages containing information about the topic - an example of a hub page is a resource list on some specific topic.
- When first introduced, HITS was able to work with the existing search engines and generate results that were comparable to hand-compiled information in directories such as Yahoo. HITS use the results of a search engine query to make its analysis query-specific.

Strengths and weaknesses of HITS

Strength :

1. Its ability to rank pages according to the query topic, which may be able to provide more relevant authority and hub pages.
2. HITS is sensitive to user query.
3. Important pages are obtained on basis of calculated authority and hubs value.

Weaknesses :

1. It is easily spammed.
2. Topic drift. Many pages in the expanded set may not be on topic.
3. Inefficiency at query time : The query time evaluation is slow.
4. Topic drift occurs when there are irrelevant pages in the root set and they are strongly connected.

8.7.3 Difference between HITS and PageRank

Sr.No.	HITS	PageRank
1.	HITS based on web structure mining and web content mining.	PageRank based on web structure mining.
2.	The most popular social networking website Twitter makes use of HITS algorithm in suggesting user accounts to follow.	Used by the Google Internet search engine uses the link structure of the web to determine the importance of the web page.

Scanned by TapScanner

3.	HITS is applied on a subgraph after a search is done on the complete graph.	PageRank is used for ranking all the nodes of the complete graph and then applying a search.
4.	HITS defines hubs and authorities recursively.	PageRank is based on the 'random surfer' idea and the web is seen as a Markov Chain.
5.	A page is a good hub if it points to good authorities, and a page is a good authority if it is pointed to by good hubs.	A web page is important if it is pointed to by other important web pages.
6.	HITS is search query dependent.	PageRank which is a static ranking algorithm.
7.	HITS algorithm is executed at the query time rather than at the time of indexing.	PageRank is calculated and stored at the time of indexing and not at the time of query.
8.	HITS algorithm makes use of both backlinks and forward links.	PageRank algorithm uses only backlinks.

8.8 Multiple Choice Questions

Q.1 **_____ is a program which browses the World Wide Web in a methodical, automated manner.**

a Web crawler
b HTML
c URL
d Web

Q.2 **_____ is the act of modifying a website to increase its ranking in organic, crawler-based listings of search engines.**

a Google Search Engine
b Search Engine Optimization
c Web crawler Optimization
d All of these

Q.3 **_____ occurs when a page contains a link such that the hypertext points to a page with no outgoing links.**

a Deep link
b Dark link
c Dangling link
d All of these

Q.4 **_____ are index pages that provide lots of useful links to relevant content pages (topic authorities).**

a Authorities
b Hub
c Web page
d None

Scanned by TapScanner

Q.5 SEO copywriting aims to produce _____ on a website.

a page
b rank
c value
d content

Q.6 Which of the following is not the factors that affect the search engine ranking ?

a Visible on-page factors
b Invisible on-page factors
c Time-based factors
d Internal factors

Q.7 _____ web pages do not change content or layout with every request to the web server.

a Dynamic
b Static
c Dynamic and static
d None

Q.8 Boolean search is a search that uses the _____ operators.

a binary
b relational
c logical
d bitwise

Q.9 _____ Link analysis is related to the statistical properties of links and individual nodes .

a Mesoscopic level
b Macroscopic level
c Microscopic level
d All of these

Q.10 The graph model in _____ link analysis is induced from two kinds of relationships, that is, block-to-page and page-to-block.

a page level
b text level
c block level
d document level

Q.11 _____ ranking methods use the query to rank all documents in the order of relevance.

a Text
b Word
c Information
d Document

Answer Keys for Multiple Choice Questions

Q.1	b	Q.2	b	Q.3	c	Q.4	b
Q.5	d	Q.6	d	Q.7	b	Q.8	c
Q.9	c	Q.10	c	Q.11	d		

Scanned by TapScanner

9 Retrieving Structured Documents

Syllabus

XML retrieval, semantic web.

Contents

9.1 XML Retrieval

9.2 Semantic Web

Scanned by TapScanner

9.1 XML Retrieval

- Documents can be structured or unstructured. Unstructured documents have no fixed pre - defined format, whereas structured documents are usually organized according to a fixed pre - defined structure.
- An example of a structured document is a book organized into chapters, each with sections made of paragraphs and so on.
- Nowadays, the most common way to format structured content is with the W3C standard for information repositories and exchanges, the eXtensible Mark-up Language (XML).
- To identify the most useful XML elements to return as answers to given queries, XML information retrieval systems require :
  1. Query languages that allow users to specify the nature of relevant components, in particular with respect to their structure.
  2. Representation strategies providing a description not only of the content of XML documents, but also their structure.
  3. Ranking strategies that determine the most relevant elements and rank these appropriately for a given query.
- XML documents are organized into a logical structure, as provided by the XML mark-up.

Query languages

- There are a number of techniques to enhance the usefulness of the queries. Some examples are the expansion of a word to the set of its synonyms or the use of a thesaurus. Some words which are very frequent and do not carry meaning may be removed. We refer to words that can be used to match query terms as keywords.
- Another issue is the subject of the retrieval unit the information retrieval system adopts. The retrieval unit is the basic element which can be retrieved as an answer to a query. We call the retrieval units simply documents, even if this reference can be used with different meanings.

9.1.1 Types of Queries

- **Keyword Based Querying :** A query is the formulation of a user information need. Keyword based queries are popular, since they are intuitive, easy to express, and allow for fast ranking. However, a query can also be a more complex combination of operations involving several words.
- **Word Queries :** The most elementary query that can be formulated in a text retrieval system is the word. Some models are also able to see the internal division of words into letters. The division of the text into words is not arbitrary, since

Scanned by TapScanner

words carry a lot of meaning in natural language. The result of word queries is the set of documents containing at least one of the words of the query. The resulting documents are ranked according to the degree of similarity with respect to the query. To support ranking, two common statistics on word occurrences inside texts are commonly used (term frequency and inverse document frequency).

- **Context Queries :** Many systems complement queries with the ability to search words in a given context. Words which appear near each other may signal higher likelihood of relevance than if they appear apart.
- **Boolean Queries :** The oldest way to combine keyword queries is to use boolean operators. A boolean query has a syntax composed of atoms and operators. Atoms are basic queries that retrieve documents and boolean operators work on their operands and deliver sets of documents.

9.1.2 Patterns Matching

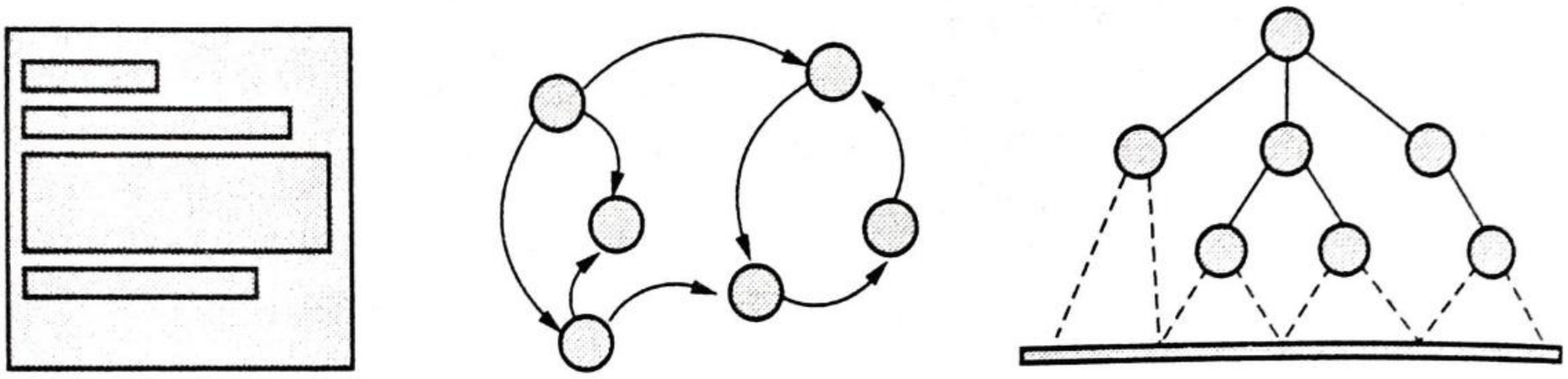
- A pattern is a set of syntactic features that must be found in a text segment. Those segments satisfying the pattern specifications are said to match the pattern.
- We can search for documents containing segments which match a given search pattern. Each system allows specifying some types of patterns. The more powerful the set of patterns allowed the more involved queries can the user formulates.
- **The most used types of patterns are :**
  1. Words : A string which must be a word in the text
  2. Prefixes : A string which must form the beginning of a text word
  3. Suffixes : A string which must form the termination of a text word
  4. Substrings : A string which can appear within a text word
  5. Ranges : A pair of strings which matches any word which lexicographically lies between them
  6. Allowing errors : A word together with an error threshold
  7. Regular expressions : A rather general pattern built up by simple strings
  8. Extended patterns : A more user-friendly query language to represent some common cases of regular expressions.

9.1.3 Structural Queries

- The text collections tend to have some structure built into them. The standardization of languages to represent structured texts has pushed forward in this direction.

Scanned by TapScanner

- Mixing contents and structure in queries allows posing very powerful queries. Queries can be expressed using containment, proximity or other restrictions on the structural element.
- The three main types of structures :
  a) Form-like fixed structure b) Hypertext structure
  c) Hierarchical structure.
- Fig. 9.1.1 shows types of structures.



(a) Form - like fixed structure (b) Hypertext structure (c) Hierarchical structure

Fig. 9.1.1

9.2 Semantic Web

- The Semantic Web is not a separate Web but an extension of the current one, in which information is given well - defined meaning, better enabling computers and people to work in - co - operation. It is the application of advanced knowledge technologies to the Web and distributed systems.
- The vision of the Semantic Web is that of a world wide distributed architecture where data and services easily interoperate.
- Semantic Web is being to be developed to overcome the following problems for current web.
  1. The Web content lacks a proper structure regarding the representation of information.
  2. Ambiguity of information resulting from poor interconnection of information.
  3. Automatic information transfer is lacking.
  4. Usability to deal with enormous number of users and content ensuring trust at all levels.
  5. Incapability of machines to understand the provided information due to lack of a universal format.

Scanned by TapScanner

9.2.1 Development of Semantic Web

- The Semantic Web is an extension of the current web in which information is given well - defined meaning, better enabling computers and people to work in co - operation.
- It is a next generation of the WWW. Information has machine - processable and machine - understandable semantics
- Not a separate Web but an augmentation of the current one. The backbone of Semantic Web are ontologies.
- Semantic Web technology provides a basis for information sharing and performing some functions with generic software, but doesn't solve all application problems. The some of the areas that must be addressed by an application using Semantic Web technology.
- The main obstacle to provide better support to Web users is that, at present , the meaning of Web content is not machine accessible.
- Although there are tools to retrieve texts, but when it comes to interpreting sentence and extracting useful information for the user, the capabilities of current software are still very limited.
- In the Semantic Web, pages not only store content as a set of unrelated words in a document, but also code their meaning and structure.
- Semantic Web is set to become the future because it makes the understanding between humans and machines easy. Semantic web design methodologies use ontology languages such as RDF, OWL to represent information internally.
- The Semantic Web has been actively promoted since by the World Wide Web Consortium, the organization that is chiefly responsible for setting technical standards on the Web.
- The core technology of the Semantic Web, logic - based languages for knowledge representation and reasoning have been developed in the research field of artificial intelligence.

9.2.2 Introduction Knowledge Representation on the Semantic Web

- Semantic Web is used to represent information on and about the current Web using formal languages that computers can process. A major feature of the Semantic Web is the ability to provide definitions for objects and types of objects that are accessable and manipulable from within the Semantic Web.
- A knowledge-based system maintains a knowledge base which stores the symbols of the computational model in form of statements about the domain, and it performs reasoning by manipulating these symbols.

Scanned by TapScanner

- Knowledge is the information about a domain that can be used to solve problems in that domain. To solve many problems requires much knowledge, and this knowledge must be represented in the computer. As part of designing a program to solve problems, we must define how the knowledge will be represented.
- A representation scheme is the form of the knowledge that is used in an agent. A representation of some piece of knowledge is the internal representation of the knowledge. A representation scheme specifies the form of the knowledge. A knowledge base is the representation of all of the knowledge that is stored by an agent.
- Knowledge representation is the method used to encode knowledge in an intelligent system's knowledge base. The object of knowledge representation is to express knowledge in computer-tractable form, such that it can be used to help intelligent system perform well. Fig 9.2.1 shows knowledge base and domain.

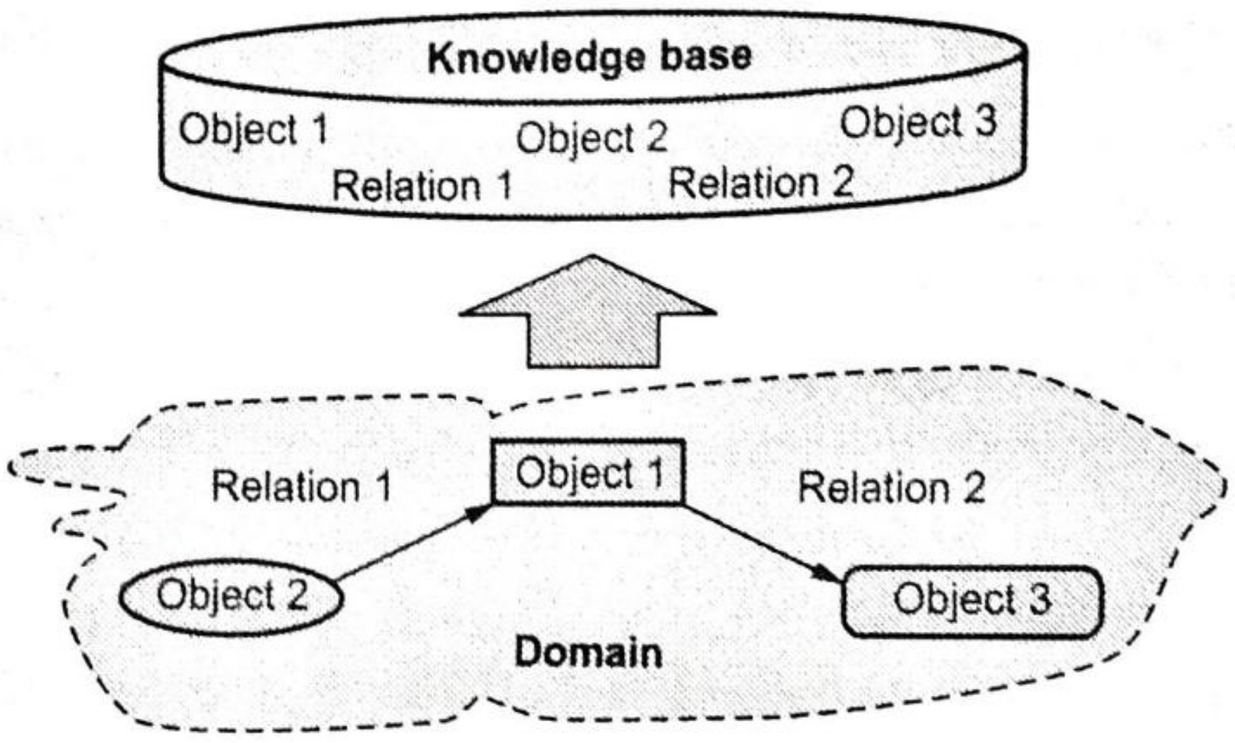


Fig. 9.2.1 : Knowledge base and domain

- A representation consists of four fundamental parts :
  1. A lexical part that determines which symbols are allowed in the representation's vocabulary.
  2. A structural part that describes constraints on how the symbols can be arranged.
  3. A procedural part that specifies access procedures that enable to create descriptions, to modify them, and to answer questions using them.
  4. A semantic part that establishes a way of associating meaning with the description.
- If we look at current Semantic Web technologies and use cases, knowledge representation appears in different forms, the most prevalent of which are based on semantic networks, rules and logic.

Scanned by TapScanner

- Semantic network structures can be found in RDF graph representations or Topic Maps, whereas a formalization of business knowledge often comes in form of rules with some "if-then" reading, e.g. in business rules or logic programming formalisms. Logic is used to realize a precise semantic interpretation for both of the other forms.
- A semantic network is a graph whose nodes represent concepts and whose arcs represent relations between these concepts. They provide a structural representation of statements about a domain of interest. In the business trips domain, typical concepts would be "Company", "Employee" or "Flight", while typical relations would be "books", "isEm-ployedAt" or "participatesIn".
- In terms of the knowledge representation and reasoning semantic web lets us :
  1. Represent the knowledge
  2. Support search queries on knowledge and matches
  3. Support inference.

9.2.3 Ontology and their Role in the Semantic Web

- The term ontology is originated from philosophy. In that context it is used as the name of a subfield of philosophy, namely, the study of the nature of existence.
- For the Semantic Web purpose : "An ontology is an explicit and formal specification of a conceptualisation".
- In general, an ontology describes formally a domain of discourse. An ontology consists of a finite list of terms and the relationships between the terms. The terms denote important concepts classes of objects of the domain.
- For example, in a university setting, staff members, students, courses, modules, lecture theatres, and schools are some important concepts.
- In the context of the Web, ontologies provide a shared understanding of a domain. Such a shared understanding is necessary to overcome the difference in terminology.
- Ontologies are useful for improving accuracy of Web searches. Web searches can exploit generalization/specialization information.
- Ontologies are the core concept of Knowledge Representation in the Semantic Web. RDFS and OWL are languages that can produce such models. The Semantic Web is build on the eXtensible Markup Language (XML), the Resource Description Framework (RDF).

Scanned by TapScanner

Ontology Languages for the Web

- RDF is a universal language that enables users to describe their own vocabularies. But, RDF does not make assumption about any particular domain. It is up to user to define this in RDF schema.
- RDF is a directed, labeled graph data format for representing information in the Web. Most forms of the query languages contain a set of triple patterns. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable.
- RDF Schema is a vocabulary description language for describing properties and classes of RDF resources, with a semantics for generalization hierarchies of such properties and classes.
- OWL is a richer vocabulary description language for describing properties and classes.
- Ontologies are domain models with two special characteristics, which lead to the notion of shared meaning or semantics :
  1. Ontologies are expressed in formal languages with a well - defined semantics.
  2. Ontologies build upon a shared understanding within a community.
- Semantic networks provide a means to abstract from natural language, representing the knowledge that is captured in text in a form more suitable for computation.
- Semantic networks are closely related to another form of knowledge representation called **frame systems.** In fact, frame systems and semantic networks can be identical in their expressiveness but use different representation metaphors.
- While the semantic network metaphor is that of a graph with concept nodes linked by relation arcs, the frame metaphor draws concepts as boxes, i.e. frames, and relations as slots inside frames that can be filled by other frames. Thus, in the frame metaphor the graph turns into nested boxes.
- The semantic network form of knowledge representation is especially suitable for capturing the taxonomic structure of categories for domain objects and for expressing general statements about the domain of interest.
- Ontologies emerged as an alternative to represent knowledge. However, they have been used to support a great variety of tasks.
- The use of ontologies to overcome the limitations of keyword - based search has been put forward as one of the motivations of the Semantic Web.

Why Develop Ontology ?

- Other reasons are as follows :
  1. To enable a machine to use the knowledge in some application.

Scanned by TapScanner

2. To enable multiple machines to share their knowledge.
3. To help yourself understand some area of knowledge better.
4. To help other people understand some area of knowledge.
5. To help people reach a consensus in their understanding of some area of knowledge.

- The majority of existing ontologies are 'simple' taxonomies or classifications, i.e., hierarchically organized categories used to classify resources.
- Ontologies with arbitrary relations do exist, but no intuitive and efficient reasoning techniques support such ontologies in general. So we need 'lightweight' ontologies.
- **Lightweight ontology** is typically applied to ontologies that make a distinction between classes, instances and properties, but contain minimal descriptions of them.
- **Heavyweight** ontologies allow to describe more precisely how classes are composed of other classes, and provide a richer set of constructs to constrain how properties can be applied to classes

9.2.4 Ontology Languages for the Semantic Web

9.2.4.1 Resource Description Framework (RDF) and RDF Schema

- The Resource Description Framework (RDF) is a W3C Recommendation since February 2004 which jointly replaced RDF Model and Syntax and RDF Schema.
- RDF provides a means for adding semantics to a document without making any assumptions about the structure of the document and it provides pre - defined modeling primitives for expressing semantics of data.
- RDF is domain - independent and can be used to model both real world objects and information resources. RDF itself is a very primitive modeling language, but it is the basis of more complex languages such as OWL.
- RDF was developed with the motivation to provide web meta data and open information models, to get new information by combining data from several applications and to enable automated processing of web information by software agents.
- RDF is the foundation layer of the Semantic Web. The semantics are encoded in sets of triples, where each triple consists of a subject, a predicate or property and an object, similar to what we have in natural language sentences.
- An RDF statement has three components : **a subject, a predicate, and an object.**
  1. The subject is the source of the edge and must be a resource.

Scanned by TapScanner

2. The object of a statement is the target of the edge. Like the subject, it can be a resource identified by a URI, but it can alternatively be a literal value like a string or a number.
3. The predicate of a statement determines what kind of relationship holds between the subject and the object. It too is identified by a URI.

- Fig. 9.2.2 shows an example RDF graph.

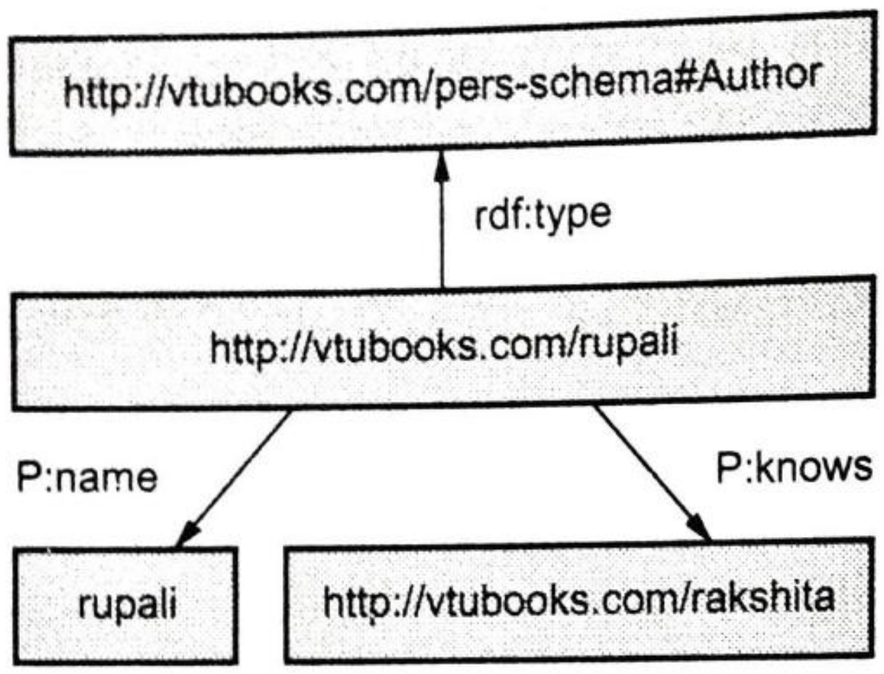


Fig. 9.2.2 : Example RDF graph

- The RDF Vocabulary Description language RDF Schema (RDFS) is an extension to RDF which facilitates the formulation of vocabularies for RDF meta data. While RDF is used to relate resources by means of properties, RDFS introduces the notions of resource classes and their hierarchies.
- The combined use of both RDF and RDFS is often referred to as RDF(S) and provides a simple ontology language for conceptual modeling with some basic interfacing capabilities.
- Fig. 9.2.3 shows an RDF graph, which is a set of RDF triples. A node may be a URI reference or blank, which means that it is a unique node with no separate form of identification. The object node can also be a literal. The property is also a URI. (See Fig. 9.2.3 on next page.)
- An RDF graph is defined as a set of RDF triples. A subgraph of an RDF graph is a subset of the triples in the graph.
- A triple is identified with the singleton set containing it, so that each triple in a graph is considered to be a subgraph.
- A proper subgraph is a proper subset of the triples in the graph. A ground RDF graph is one with no blank nodes.

Scanned by TapScanner

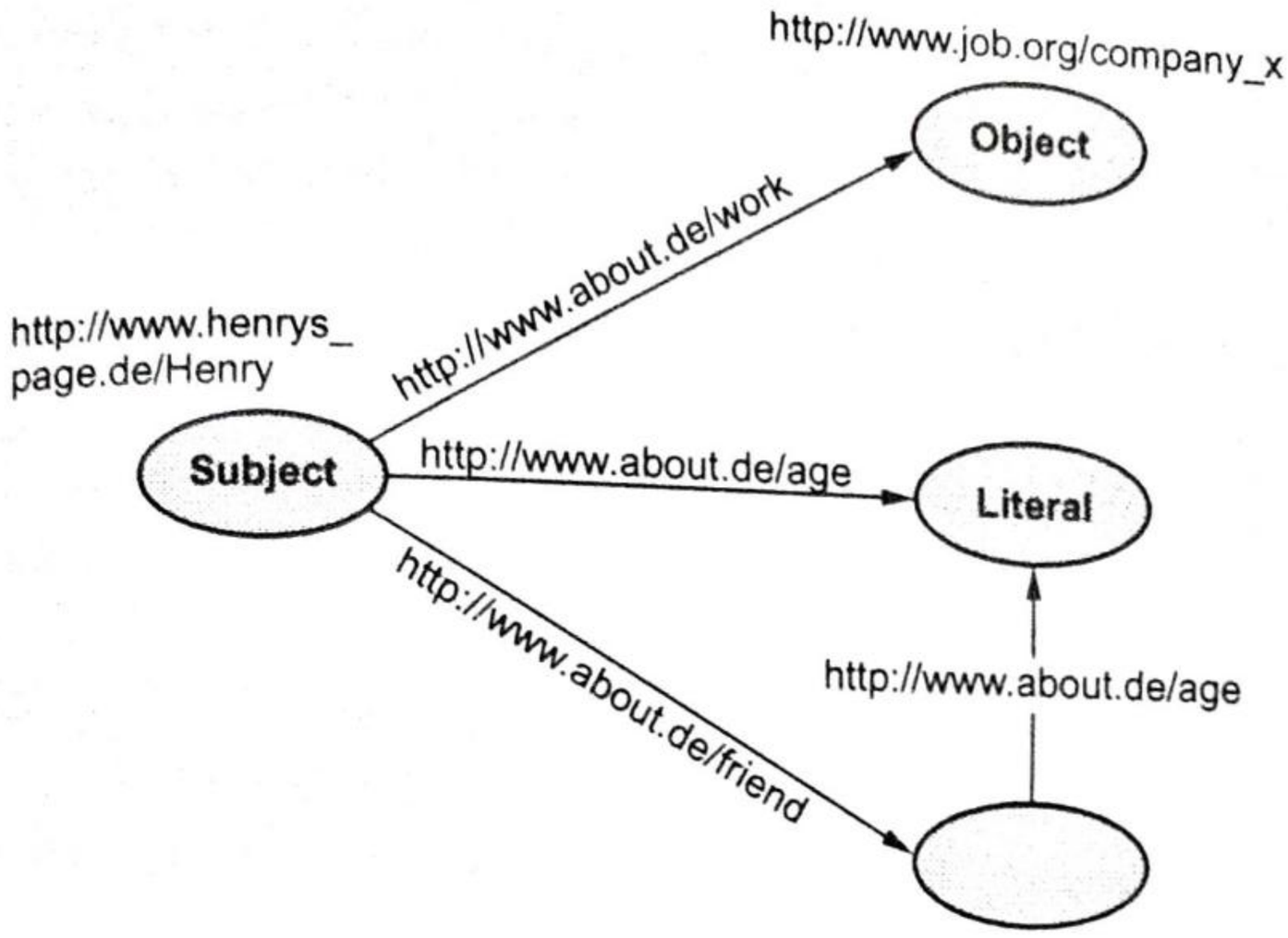


Fig. 9.2.3 : RDF triple relations between resources, literals and blank - nodes

- The Resource Description Language Schema (RDFS) is a vocabulary language that provides the users to define terms they intend to use in their RDF document, similar to the design in Object Oriented Programming (OOP) languages.
- A set of names is referred to as a vocabulary. The vocabulary of a graph is the set of names which occur as the subject, predicate or object of any triple in the graph.
- The four important RDFS vocabulary definitions are **rdfs:Class, rdf:Property, rdfs:domain and rdfs:range**. They define which nodes are connected through a certain property.
- Things described by RDF expressions are called resources, and are considered to be instances of the class rdfs:Resource. The RDF class rdfs:Resource represents the set called 'Resources' in the formal model.
  1. **rdf:Property :** This represents the subset of RDF resources that are properties, i.e., all the elements of the set called 'Properties' the formal model. Example : ex:author rdf:type rdf:Property .
- Then, property ex:author can be used as a **predicate in an RDF triple** such as the following :

  ex:Rupali ex:sportman ex:cricket
- In RDFS property definitions are independent of class definitions. In other words, a property definition can be made without any reference to a class.
- Optionally, properties can be declared to apply to certain instances of classes by defining their domain and range.

Scanned by TapScanner

2. **rdfs:Class** : This corresponds to the generic concept of a Type or Category, similar to the notion of a Class in object - oriented programming languages such as Java. When a schema defines a new class, the resource representing that class must have an rdf:type property whose value is the resource rdfs:Class. RDF classes can be defined to represent almost anything, such as Web pages, people, document types, databases or abstract concepts.

- Every RDF model which uses the schema mechanism also (implicitly) includes the core properties. These are instances of the rdf:Property class and provide a mechanism for expressing relationships between classes and their instances or superclasses.

3. **rdf:type** : This indicates that a resource is a member of a class, and thus has all the characteristics that are to be expected of a member of that class. The value of an rdf:type property for some resource is another resource which must be an instance of rdfs:Class. The resource known as rdfs:Class is itself a resource of rdf:type rdfs:Class.

4. **rdfs:range** : An instance of ConstraintProperty that is used to indicate the class(es) that the values of a property must be members of. The value of a range property is always a Class. Range constraints are only applied to properties. A property can have at most one range property. It is possible for it to have no range, in which case the class of the property value is unconstrained.

5. **rdfs:domain** : This is an instance of ConstraintProperty that is used to indicate the class(es) on whose members a property can be used. If a property has no domain property, it may be used with any resource. If it has exactly one domain property, it may only be used on instances of that class. If it has more than one domain property, the constrained property can be used with instances of any of those classes.

- For a property, we can have zero, one, or more than one domain or range statements. **No domain or no range statement** : If no range statement has been made for property P, then **nothing has been said about the values** of this property. Similarly for no domain statement.

Some Utility Properties

1. rdfs:label
2. rdfs:comment
3. rdfs:seeAlso
4. rdfs:isDefinedBy.

Scanned by TapScanner

- **rdfs:label** is an instance of rdf:Property that may be used to provide a human-readable version of a resource's name. The rdfs:domain of rdfs:label is rdfs:Resource. The rdfs:range of rdfs:label is rdfs:Literal. Multilingual labels are supported using the language tagging facility of RDF literals.
- **rdfs:comment** is an instance of rdf:Property that may be used to provide a **human-readable description of a resource**. The rdfs:domain of rdfs:comment is rdfs:Resource. The rdfs:range of rdfs:comment is rdfs:Literal. Multilingual documentation is supported through use of the **language tagging** facility of RDF literals.
- **rdfs:seeAlso** is an instance of rdf:Property that is used to indicate a resource that might provide additional information about the subject resource. A triple of the form S rdfs:seeAlso O states that the resource O may provide additional information about S. It may be possible to retrieve representations of O from the Web, but this is not required. When such representations may be retrieved, no constraints are placed on the format of those representations.
- The rdfs:domain of rdfs:seeAlso is rdfs:Resource. The rdfs:range of rdfs:seeAlso is rdfs:Resource.
- **rdfs:isDefinedBy** is an instance of rdf:Property that is used to indicate a resource defining the subject resource. This property may be used to indicate an RDF vocabulary in which a resource is described.
- RDF Schema descriptions are not prescriptive in the way programming language type declarations typically are.
- RDF Schema provides schema information as additional descriptions of resources, but does not prescribe how these descriptions should be used by an application.
- RDF is a graph data model. RDF data are directed, labeled graphs. A single edge in an RDF graph is a 3-tuple that is called either a statement or triple.
- Triples are organized into named graphs, forming 4-tuples, or quads. RDF resources (nodes), predicates (edges), and named graphs are labeled by URIs.
- The RDFS language is used by various web-based applications for describing meta data, and a number of tools are available that support visual editing and programmatic handling of RDFS descriptions.
- RDF is a language for expressing the information that needs to be processed by applications, so that it can be exchanged without loss of meaning. The data does not need to be stored in RDF but can be created on the fly from relational databases or other non-RDF sources.

Scanned by TapScanner

RDF vocabulary :

Basic constructs	1.	rdf:type
	2.	rdf:Property
	3.	rdf:XMLLiteral
Collections	1.	rdf:List
	2.	rdf:Seq
	3.	rdf:Bag
	4.	rdf:Alt
	5.	rdf:first
	6.	rdf:rest
	7.	rdf:nil
	8.	rdf:_n
	9.	rdf:value
Reification	1.	rdf:Statement
	2.	rdf:subject
	3.	rdf:predicate
	4.	rdf:object

RDF Schema vocabulary

Basic constructs	1.	rdfs:domain
	2.	rdfs:range
	3.	rdfs:Resource
	4.	rdfs:Literal
	5.	rdfs:Datatype
	6.	rdfs:Class
	7.	rdfs:subClassOf
	8.	rdfs:subPropertyOf

Scanned by TapScanner

Collections	1.	rdfs:member
	2.	rdfs:Container
	3.	rdfs:ContainerMembershipProperty
Documentation & reference	1.	rdfs:comment
	2.	rdfs:seeAlso
	3.	rdfs:isDefinedBy
	4.	rdfs:label

9.2.4.2 Web Ontology Language (OWL)

- The Web Ontology Language (OWL) is a knowledge representation language designed by the W3C Web Ontology Working Group for use by applications that need to process web content.
- The OWL standard defines different syntaxes based on RDF(S), XML and proprietary text format. The OWL RDF/XML syntax allows for an encoding of an OWL ontology within the RDF(S) framework in RDF/XML serialisation. The OWL XML presentation syntax pro-vides a more compact XML format for OWL ontologies, independent from RDF(S).
- OWL can also define two types of properties : **object properties and datatype properties.**
- Object properties specify relationships between pairs of resources. Datatype properties, on the other hand, specify a relation between a resource and a data type value; they are equivalent to the notion of attributes in some formalisms.
- OWL Lite, OWL DL and OWL Full are the three levels of OWL.
- OWL Lite supports simple constraints and classification hierarchies which are used to construct taxonomies and thesauri. The cardinality restriction is limited to 0 and 1.
- OWL DL, where DL is the abbreviation for Description Logic, provides maximum expressiveness while retaining computational completeness and decidability.
- Unlike OWL DL, OWL Full does not guarantee computational completeness. Classes described in OWL Full can be treated simultaneously as instances of collections.
- An OWL document can contain the following declarations :
  1. Header
  2. Classes
  3. Complex classes
  4. Individuals

Scanned by TapScanner

5. Properties, property characteristics and property restrictions
6. Ontology mapping

OWL Full vocabulary

Boolean class combinations	1. owl:unionOf
	2. owl:complementOf
	3. owl:intersectionOf
(In)equality of classes and instances	1. owl:equivalentClass
	2. owl:disjointWith
	3. owl:equivalentProperty
	4. owl:sameAs
	5. owl:differentFrom
	6. owl:AllDifferent
	7. owl:distinctMembers
Enumerated types	1. owl:oneOf
	2. owl:DataRange
Property characteristics	1. owl:ObjectProperty
	2. owl:DatatypeProperty
	3. owl:inverseOf
	4. owl:TransitiveProperty
	5. owl:SymmetricProperty
	6. owl:FunctionalProperty
	7. owl:InverseFunctionalProperty

Scanned by TapScanner

Property restrictions	1. owl:Restriction
	2. owl:onProperty
	3. owl:allValuesFrom
	4. owl:someValuesFrom
	5. owl:hasValue
	6. owl:minCardinality
	7. owl:maxCardinality
	8. owl:cardinality
Ontology versioning	1. owl:versionInfo
	2. owl:priorVersion
	3. owl:backwardCompatibleWith
	4. owl:incompatibleWith
	5. owl:DeprecatedClass
	6. owl:DeprecatedProperty
Ontology metadata	1. owl:Ontology
	2. owl:imports
	3. owl:AnnotationProperty
	4. owl:OntologyProperty

❑❑❑

Scanned by TapScanner