Experiment No. Assignment No-2 Date :

Q.1 List & explain various types of queries in information retrieval ?

→ Queries in IR

- → Keyword Based Querying
- → Pattern Matching
- → structural Queries
- → Query protocols.

1. Keyword Based Querying :-

- A query is the expansion of a user's information requirement.

(A) Single-Word Queries :-

(1) A word is normally defined in a straightforward manner.
(2) A word is a sequence of letters surrounded by separators and the alphabet is divided into letters & seperators.
(3) The set of documents containing at least one of the query's words is the result of word queries.
(4) Two common statistics on word occurrences are ranked based on their similarity to the query.

(B) Context Queries

1) Many systems supplement single-word queries with the

ability to search for words in a given context that is, near the words.
(2) Words that appear close together may indicate a higher likelihood of relevance than words that appear separately.
(3) For example, we may want to form phrases of words or words that are close together in the text.

(C) Boolean-Queries :-

1) A boolean query has a syntax that is made up of atoms that retrieve documents & Boolean operators that work on their operands & return a set of documents
2) This is similar to the syntax trees of arithmetic expressions, where the leaves are numbers & variables and the internal nodes are operations.

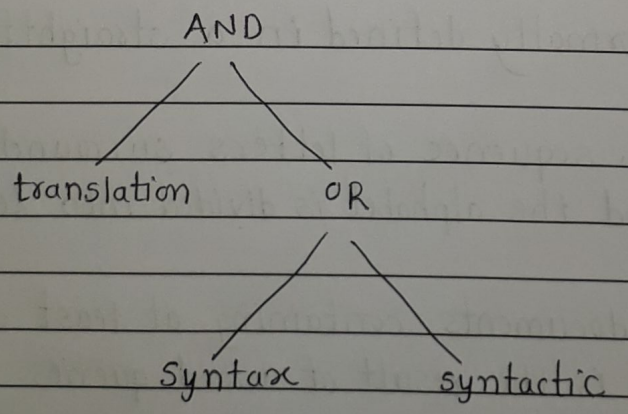


Fig: An example of a query syntax tree.

D) Natural Language :-

1) By pushing the fuzzy boolean model even further, the distinction between AND and OR could be completely blurred.
2) The negation can be handled by allowing the user to express that certain words are undesirable, in which the documents that contain them are penalised in the ranking computation.
3) A threshold can be set to prevent documents with extremely low weights from being retrieved.

2. Pattern Matching :-

- These data retrieval queries can be applied to linguistic text statistics & data extraction.
- Their output can be fed into the composition mechanism to form phrases & proximity queries, which comprise what we call basic queries.

The most common patterns are :-

1) Words :-
a string which must be a word in the text. This is the most basic pattern.

2) Prefixes :-
a string which must form the beginning of a text word.

3) suffix :-
a string which must form the termination of a text word.

4) Substrings :
a string which can appear within a text word.

5) Ranges :-
are a pair of strings that match any word that is lexicographically between them.

6) Allowing for errors :-
a word plus an error threshold. This search pattern returns all text words that are similar to the specified word.

3. Structural Queries :-

- Combining contents & structure in queries enables the creation of extremely powerful queries that are far more expressive than each query mechanism alone.
- The retrieval quality of textual database can be improved by using a query language that integrates both types of queries.
- In addition, some structural constraints can be expressed in the documents through containment, proximity or other restrictions on structural elements.

(A) Fixed structure :-

- Text structure has traditionally been quite restricted The documents like a filled form had a fixed set of fields.

(B) Hypertext

- In terms of structuring power, hypertext most likely represents inverse trend.
- A hypertext is a directed graph in which the node contains text and the links represent connections between nodes or positions within nodes.

(C) Hierarchical structures :-

- An intermediate structuring model which lies between fixed structure & hypertext is the hierarchical structure.
- This represents a recursive decomposition of the text and it is natural model for many text collections

4. Query Protocols :-

1. Z39.50

- It is a protocol that ANSI & NISO approved as a standard in 1995.
- This database is assumed to be a text collection with

some fixed fields.

2) WAIS

- (Wide Area Information service) is a set of protocols that was popular in the early 1990s before the WWW boom.
- There are several query protocol proposals in the CD-Rom publishing area.

3) CCL

- (common command Language) is a NISO proposed based on Z39.50
- It specifies 19 commands that can be executed interactively

4) CD-RDx

- (common Disk Read Only Data Exchange) employs a client server architecture and is supported by the majority of platforms.

5) SFQL

- (structured Full-text Query Language) is a client-server architecture that is based on SQL.
- Documents are relational table rows that can be tagged with GSML.

Q.2 Explain pattern matching ? What are the different types of pattern used for pattern matching ?

→ • These data retrieval queries can be applied to linguistics text statistics and data extraction
• Their output can be fed into the mentioned composition mechanism to form phrases & proximity queries, which comprise what we call basic queries.
• Each system allows to specify different types of patterns, ranging from very simple to quite complex.

The most common patterns are :-

1) Words :-
a string which must be a word in the text word. This is the most basic pattern.

2) Prefixes :-
A string which must form the beginning of a text word.

3) Suffixes :-
A string which must form a termination of a text word.

4) Substrings :-
A string which can appear within a text word.

5) Ranges :- Are a pair of strings that match any word that is lexicographically between them.

vi) Allowing for errors :-

A word plus an error threshold.

This search pattern returns all text words that are "similar" to the specific word. The concept of similarity can be defined in a variety of ways.

3. Explain Structural Queries in detail.

→ Combining contents and structures in queries enables the creation of extemely powerful queries that are far more expressive than each query mechanism alone.

- The retrieval quality of textual database can be improved by using a query language that integrates both types of queries.

(A) Fixed Structure :-

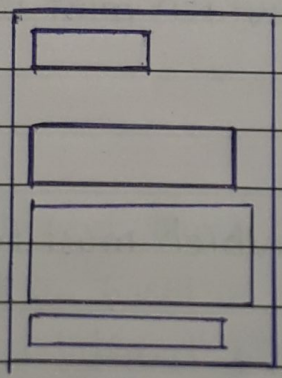


fig: form-like fixed structure

- Text structure has traditionally been quite restricted. The documents like a filled form, had a fixed set of fields.

- Each field contains some text. Some fields were not present in all documents, but they could rarely appear in any order or repeatedly across the document.

(B) Hypertext

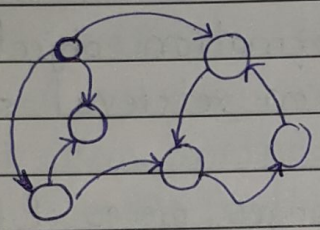


Fig :- Hypertext structure

- In terms of structuring power, hypertext most likely represent the inverse trend.
- A hypertext is a directed graph in which node contains text & the links represents connections between nodes or position within nodes.

(C) Hierarchical structure :-

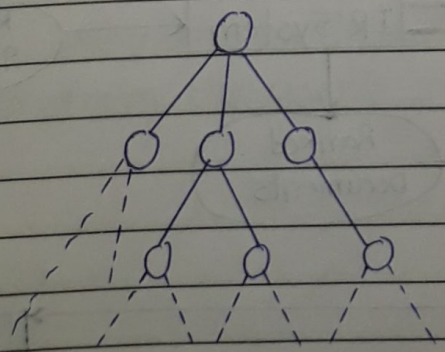


fig : Hierarchical structure

- An intermediate structuring model which lies between fixed structure and hypertext is the hierarchical structure.
- This represents a recursive decomposition of the text & it is a natural model for many text collections.

Q.4 Explain the user relevance feedback in detail.

→
- A large number of information objects are stored electronically & information retrieval systems allow users to access them.
- These objects can be images, pieces of texts, webpages or segments of video.
- In IR system process relevance refers to how well the retrieved documents or set of documents meet the user's information needs.
- Relevance feedback is very effective for retrieval accuracy. It is an iterative process that helps to improve the retrieval systems performance & accuracy.

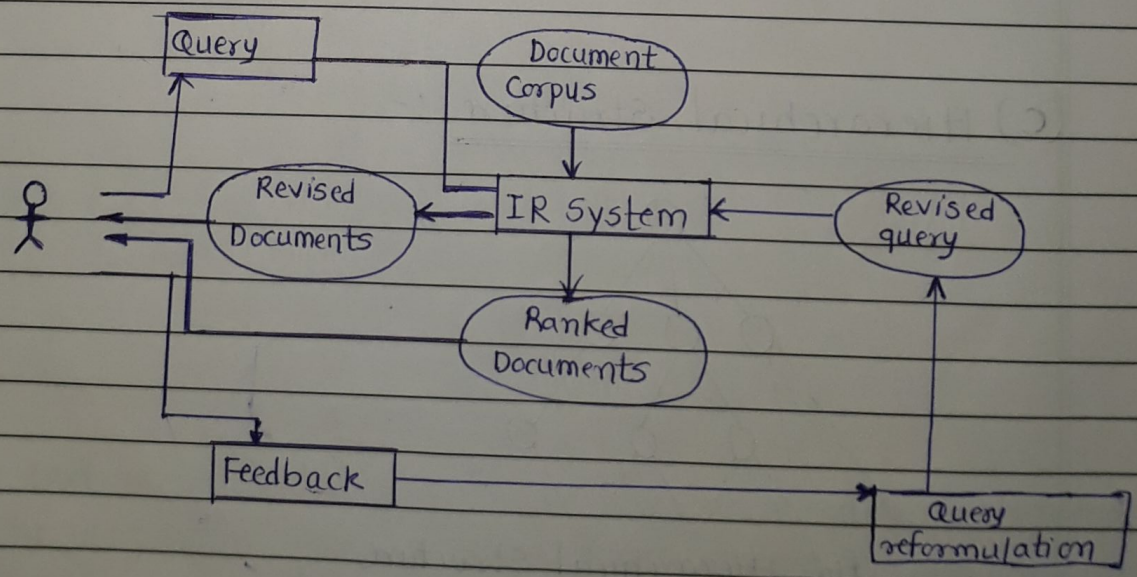


Fig 1- Relevance feedback process.

* The user relevance feedback is represented in two forms
  1) Explicit feedback
  2) Implicit feedback

Explicit feedback :-

- The user indicates the relevance of the documents returned by a query.
- There are two methods for indicating relevance.
- Binary relevance feedback indicates whether or not the document is relevant.
- The importance of document is indicated by providing some kind of description, either through words or through numerical scaling.

Implicit feedback :-

- Noticing user behaviour indicates this type of feedback.
- The amount of time spent viewing a document or browsing can be used to provide relevant information to the user.
- That is why it is called Implicit feedback ; it takes feedback from the user's actions implicitely.

Q.5 Discuss about local analysis versus global analysis.

→ ◦ Local : Documents retrieved are examined to automatically determine query expansion. No relevance feedback needed.

◦ Global :- Thesaurus used to help select terms for expansion.

* Thesaurus :-

A thesaurus provided information on synonyms & semantically related words & phrases.

Eg :-

physician

syn : || croaker, doc, doctor, MD medical, medicine, medico, || sawbones

rel : medic, general, practitioner, surgeon

Automatic local Analysis :-

- At any time, dynamically determine similar terms based on analysis of top-ranked retrieved documents.
- Base correlation analysis on only the "local" set of retrieved documents for a specific query.
- Avoids ambiguity by determining similar terms only within relevant documents.

$$C_{if} = \sum_{d_j \in D_j} f_{sik} \times f_{sjk}$$

f_{ik} : Frequency of term i in document k.

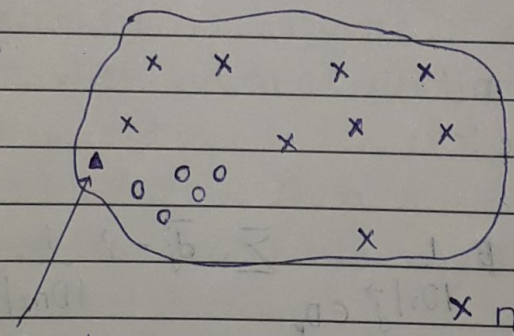
Automatic Global Analysis :-

- Determine term similarity through a pre-computed statistical analysis of the complete corpus.
- Compute association matrics which quantify term correlation in terms of the complete corpus
- Expand queries with statistically most similar terms.

Q.6 Expand the Racchio algorithm for relevance feedback.

→

The Racchio algorithm is the classic algorithm for implementing relevance feedback. It models a way of incorporating relevance feedback information into the vector space model.



Underlying Theory :-

- We want to find a query vector, denoted as $\vec{q}$, that maximizes similarity with relevant documents while minimizing similarity with non-relevant documents. If Cr is the set of relevant documents & Cnr is the set of non-relevant documents, then we wish to find : V

$$\vec{q}_{opt} = \arg \vec{q} \max \left[sim(\vec{q}, C_r) - sim(\vec{q}, C_{nr}) \right] \quad \text{---(1)}$$

- Where sim is defined as in eqn (2). Under cosine similarity, the optimal query vector $\vec{q}_{opt}$ for separating the relevant & non-relevant doc. is

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j$$

In othe words, the optimal query is the vector difference between the centroids of the relevant & non-relevant document.

Racchio Algorithm :-
Used in practice :-

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

where,

D_r = set of known relevant doc vectors
D_{nr} = set of known irrelevant doc vectors

Different from C_r & C_{nr}
q_m = modified query vector
q_0 = original query vector
α, β, γ weights.

Q.7 → Explain inverted files concept with suitable examples.

An inverted file is a data structure that maps contents to its location within a database file, a document or a collection of documents.

- It is typically composed of
  - a vocabulary containing all of the distinct words found in a text &
  - a list containing statistics about the occurrences of 't' in the text of each word t of the vocabulary
- This type of list is known as the inverted list of t.
- The most common data structure used in document retrieval system to support full text search is the inverted file.
- The inverted file index concept is, follows,
- Each document is assigned a set of keywords or attributes, with optional relevance weights assigned to each keyword.
- An inverted file is then a sorted list of keywords with each keyword containing links to the documents containing that keyword
- Using an inverted file improves search efficiency by several orders of magnitude, which is essential for very large text files.
- The cost of this efficiency is the need to store a data structure that is 10% to 100% or more the size of the text itself as well as the need to update that index as the data set changes.

Example

Words	Document
ant	doc1
demo	doc2
world	doc1, doc3

Q.8 Explain sequential searching mechanism in brief.

→
- Searching - Information retrieval is one of the most important application of computers.

Eg. Looking for a name by giving the telephone number
- We give one piece of information (KEY) and we are asked to find a record that contains other information associated with the key.
- Searching for the keys that locate records is the most time consuming action in program.
- Two types of Searching are there

1) External searching in external storage devices
2) Internal searching within the computer memory.

* Sequential Search :-

- start at the first piece of data & look at it and if it no then keeping going until you find what you are looking for or until you have reached the last.
- Eg: Analysis :- When

(I) It is useful for limited data set as it is simple and does not require data to be structured in any way.

(2) When the data to be searched is constantly changing.

Advantage :- addition / deletion is easy.

Disadvantage :- time consuming (large list)

Q.9 → Explain the weighted zone scoring concept in details.

1) Given a boolean query q and a document d, weighted zone scoring assigns to the pair (q, d) a score in the interval [0, 1] by computing a linear combination of zone scores, where each zone of the document contributes a Boolean value.

2) More specifically, consider set of documents each of which has l zones. Let $g_1, \ldots, g_l \in [0,1]$ such that $\sum_{i=1}^{l} g_i = 1$. For $1 \leq i \leq l$, let s_i be a Boolean score denoting a match between q and the i^{th} zone.

3) For instance the Boolean score from a zone could be 1 if all the query terms occur in that zone and zero otherwise; indeed, it could be any boolean function that maps the presence of query terms in a zone to 0,1 Then the weighted zone score is defined to be

$$\sum_{i=1}^{l} g_i s_i$$

Weighted zone scoring is sometimes referred to also

as ranked Boolean retrieval.

Q.10 How to calculate efficient scoring & ranking.

→ • We begin by recapping the algorithm for a query such as q = jealous gossip, two observations are immediate :-

1. The unit vector $\vec{v}(q)$ has only two non-zero components.
2. In the absence of any weighting for query terms, these non-zero components are equal - in this case both equal 0.707.

• For the purpose of ranking the documents matching this query, we are really interested in the relative scores of the documents in the collection.

• To this end, it suffers to compute the cosine similarity from each document unit vector $\vec{v}(d)$ to $\vec{v}(q)$, rather than to the unit vector $\vec{v}(q)$

For any two documents d_1, d_2.

$$\vec{v}(q) \cdot \vec{v}(d_1) > \vec{v}(q) \cdot \vec{v}(d_2) \leftrightarrow \vec{v}(q) \cdot \vec{v}(d_1) > \vec{v}(q) \cdot \vec{v}(d_2)$$

• For any document d, the cosine similarity $\vec{v}(q) \cdot \vec{v}(d)$ is the weighted sum, over all terms in the query q of the weights of those terms in d.

• We walk through the postings in the inverted index for the terms in q, accumulating the total score for each document - very much as in processing a Boolean query, except we assign a positive score to each document that appears in any of the postings being traversed.

Q.11 Explain the evaluation of unranked retrieval sets takes place.

→ 1) Precision and recall are the two most common & fundamental measures of information retrieval effectiveness.

2) These are defined first for the simple case in which an IR system returns a set of documents in response to a query

Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})}$$

$$= P(\text{relevant} \mid \text{retrieved})$$

$$\text{Recall} = \frac{\#(\text{relevant items received})}{\#(\text{relevant items})}$$

$$= P(\text{retrieved} \mid \text{relevant})$$

- These notions can be made clear by examining the following contigency table.

	Relevant	Non-relevant
Retrieved	true positive (tp)	false positive (fp)
Non-retrieved	false negative	true negative (tn)

Then $P = t_p / (t_p + f_p)$

$R = t_p / (t_p + f_n)$

$\text{Accuracy} = (t_p + f_p)(t_p + f_p + f_n + t_n)$

In general, we want some recall while tolerating a certain percentages of false positive

- Single measures that trades off precision versus recall is the f measures, which is the weighted harmonic mean of precision & recall.

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad \text{where } \beta^2 = \frac{1-\alpha}{\alpha}$$

where $\alpha \in [0,1]$ & thus $\beta^2 \in [0,\infty]$

$$F_{\beta=1} = \frac{2PR}{P+R}$$

Q.12 Explain the evaluation of the ranked retrieval process in detail.

→ Set based measures include precision recall and the f measure. They are derived from unordered set of documents.

2) If we are to evaluate the ranked retrieval results that are now standard with search engines. We must extend these measures.

3) Top k retrieved documents naturally provide appropriate set of retrieved document in a ranked retrieval context.

4) For each $(k+1)^{th}$ such set, precision and recall values can be plotted to give a precision recall curve.

5) For a single information need, average precision is the average of the precision value obtained for the set of top k ranked document existing after each relevant document is retrieved. & this value is then average over information need.

6) That is, if the set of relevant documents for an information need $q_j \in Q$ is $\{d_1, \ldots d_{mj}\}$

and R_{jk} is the set of ranked retrieval result from the top result-until you get to document d_k then

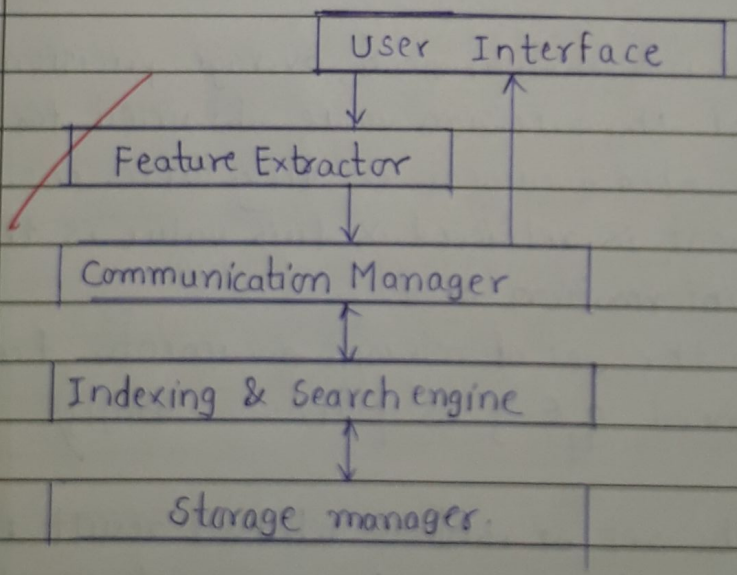
$$MAP(Q) = \frac{1}{|Q|} \sum_{j=y}^{|Q|} \frac{1}{M_j} \sum_{K=1}^{m_j} \text{precision}(R_{ik})$$

Q-13 Draw & explain architecture of multimedia information retrieval system (MIRS)

→ 1) The requirement of MIRS architecture as follows
- should be flexible and extensible to support diverse applications, query types and contents.

2) MIRS's consists
- Functional modules - added to extend deleted or placed

3) Another characteristics of MIRS is that they are normally distributed.
i) Consisting of services & clients
ii) Results from the large size of multimedia data.



4) MIRS architecture consists of

(i) User Interface :- Insertion of new multimedia items & retrieval.

(ii) Feature Extractor :- These items can be stored files or input from varient devices.

(iii) Communication Manager :- The contents of features of multimedia items are extracted either automatically.

(iv) Indexing & Search engine :- At the servers, the features are organised according to a certain indexing scheme for efficient retrieval.

(v) storage Manager :- The indexing information & the original items are stored.

Q.14 Discuss the architecture of Distributed IR? List & explain different types of DIR systems.

→ 1) A distributed system is a collection of independent computers that appears to its users as a single coherent system.

2) The first distributed system - IBM 1961 develop a compatible time sharing system. In 1972 APRANET - building blocks of the Internet was invented. The WWW concept was designed in 1989 at CERN wide spread in the 90's.

3) The ORACLE - Distributed Database Management System, Air Traffic control system - Real-time Distributed system, University Network - Client server systems are invented.

4) Distributed computing concept help information retrieval systems. Distributed IR depends on centralised IR - tries to emulate it.

Goals of Distributed computing :-

- share & access resources
- Scalability
- Transparency
- Fault Tolerance

User ⇄ Application ⇄ Distributed Database

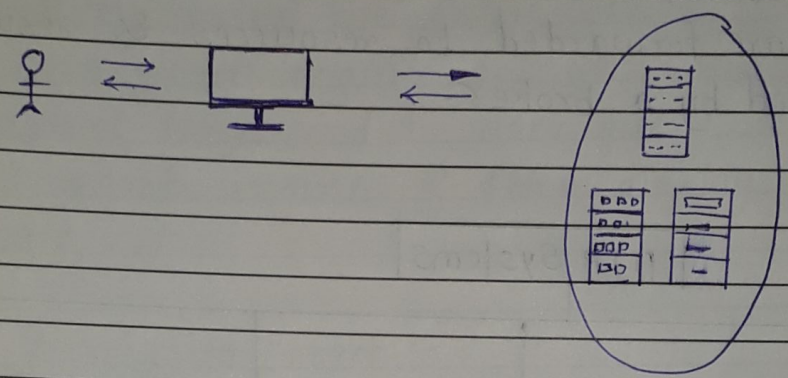


fig: Distributed IR envirnment.

* DIR Architecture :-

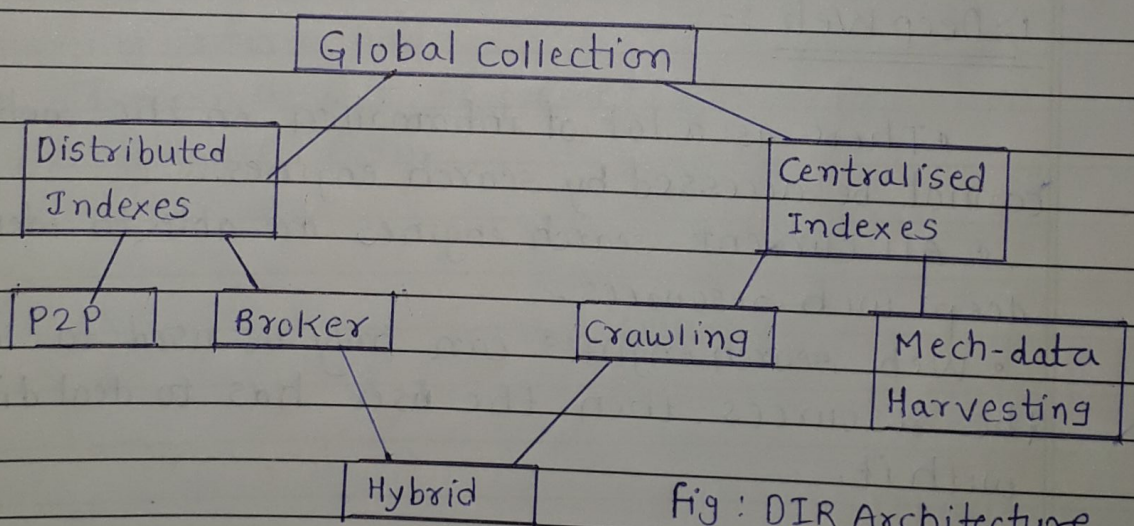


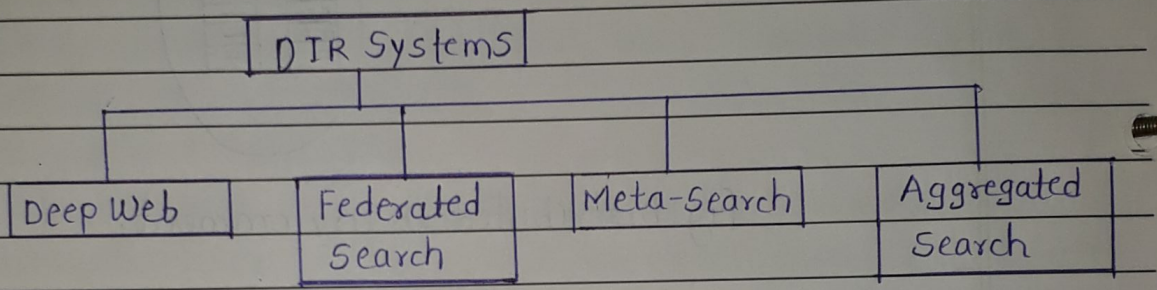
fig : DIR Architecture

1) Peer-to-Peer Network :-

- Indexes are located with the resources.
- Some part of the indexes are distributed to other resources.

2) Broker- Based Architecture :-

- Indexes are located with the resources.
- Queries are forwarded to resources & results are merged by a broker.



1) Deep Web :-

- There is a lot of information on the web that cannot be accessed by search engines.
- All current search engines are able to identify deep web resources.
- Web search engines can only be used to identify the resources, then the user has to deal directly with it.

2) Federated search :-

- DIR is also known as Federated Search
- Federated search systems do not crawl a resource but rather route a user query to the resources search facilities.

3. Metasearch :-

- Even the most powerful search engine cannot effectively crawl the entire web.
- Metasearch engines do not crawl the web; Instead they send a user query to a number of search engines & then display the merged result set.

4. Aggregated Search :-

- There is frequently more than one type of information relevant to a query (eg. web page, images, map)
- Separate indices & ranking are used for this type of information.
- It is preferable to present this information in aggregate form.

10/10/2022