Unit-I

~~Information~~

Introduction to Information Retrieval systems

- Defination of IRS
- objectives of IRS
- functional overview
- Relationship to Database management systems
- Digital libraries & Data ware houses

Information Retrieval system capabilites

- Search capabilities
- Browse capabilities
- miscellaneous capabilites.

Defination of IRS

An Information Reterival system (IRS) is a system designed to store, manage, search and retrieve relevant information for large collections of data. The Information can be in the form of text, images, audio, video or multimedia content.

ex:- Google search, Library Databases

1. Main purpose
   - Helps users find relevant information quickly
   - stores large amounts of data and retrieves it when needed

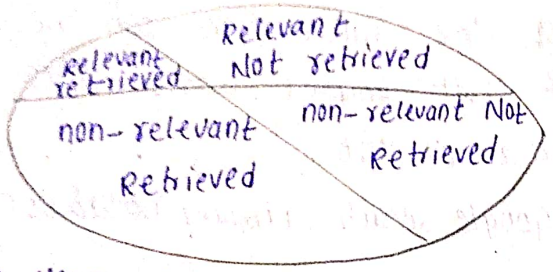
2. How IRS works
   - stores information in a structured way
   - Uses indexing to organize data for fast retrieval
   - Uses search algo to find the best results for user queries.

Key difference:

- Data Retrieval gives exact results based on structured queries.
- Information Retrieval find relevant information from unstructured data using ranking algorithms.

Objectives of Information Retrieval systems (IRS)

The primary objective of an IRS is to help users efficiently locate and retrieve relevant information while minimizing the effort required.



1. Minimizing User Overhead
   - Reducing the time and effort spent on query generation, execution, scanning results, and reviewing non-relevant items.
   - Ensuring users can quickly access the most useful information.
2. Maximizing Precision and Recall

a. Precision

Precision measures how many of the retrieved documents are actually relevant to the users query

$$\text{Precision} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total No. of Retrieved Documents}}$$

Ex:-
- Suppose a search query retrieves 50 documents out of which 30 are relevant.
- precision = 30/50 = 0.6 (or 60%.)
- This means 40% of the retrieved documents were irrelevant.

b. Recall

Recall measures how many of the total relevant documents in the database were retrieved.

$$\text{Recall:-} \frac{\text{Number of Relevant documents Retrieved}}{\text{Total No. of Relevant Documents in the Database}}$$

Ex:-
- Suppose there are 100 relevant documents in total, but only 30 were retrieved by the system.
- Recall = 30/100 = 0.3 (or 30%.)
- This means 70% of relevant documents were missed, reducing the completeness of search results.

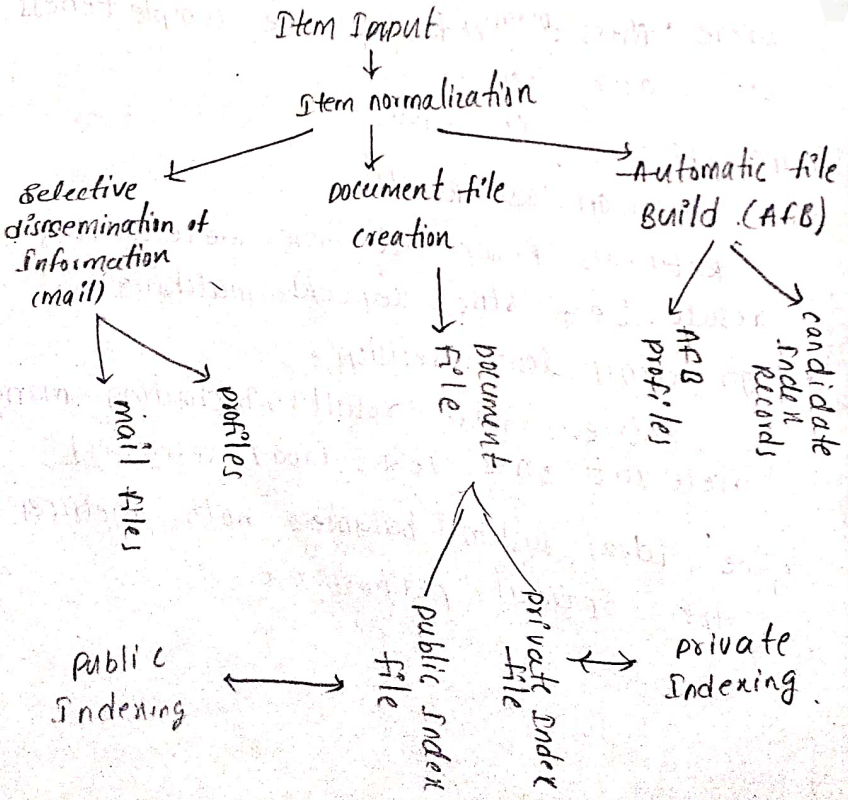
Trade-off :-
- High precision, low Recall:
  Retrieves fewer but more ~~no~~relevant results (e.g strict keyword matching)
- High Recall, low precision:
  Retrieves more results, including many irrelevant ones (e.g, broad searches)
- The ideal system balances both metrices for optimal performance

Functional overview

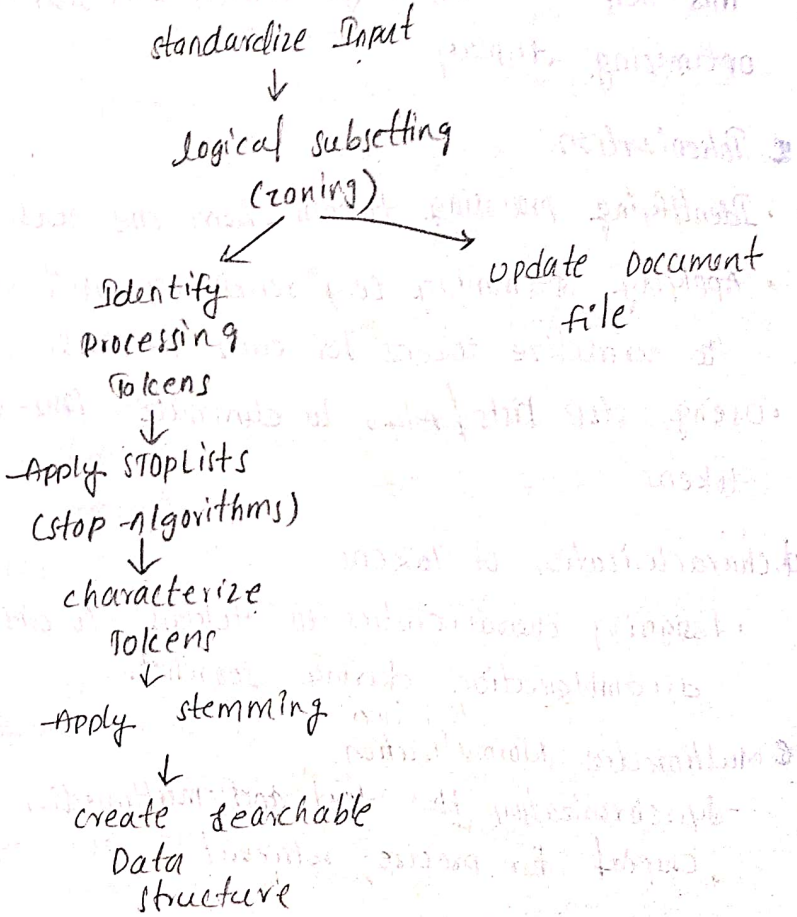
An Information Retrieval System (IRS) is designed to help users find relevant information efficiently from a large collection of data. The system processes queries, searches the database & retrieves results based on certain criteria.

It comprises four major functional process

1. Item Normalization
2. Selective Dissemination of Information
3. Document database search
4. Index Database search (including AFB process)



1. Item Normalization



Item Normalization is the first step in an integrated IRS. it involves converting incoming items into a standardized format to ensure consistency and ease of processing.

This process involves:

a. Standardization of Input formats:-
   Translating various external formats into a single, consistent data structure

b. logical restructing:- Dividing items into logical subdivisions through a process called zoning.

This helps in improving search precision and optimizing display.

c. Tokenization:
- Identifying processing tokens from the text.
- Applying stemming (e.g removing word endings) to normalize tokens for each purposes.
- Using stop lists / Algo to eliminate low-value tokens

d. Characterization of Tokens
- Assigning characteristics to tokens to aid in disambiguation during searches.

e. Multimedia Normalization.
- Synchronization blw text and multimedia is crucial for precise retrieval.

2. Selective dissimination of Information
- It is also known as mail process. The main purpose of this process is to perform comparision blw recently recieved items and the existing item statements which are required by user

key components
- search process
- mail files
- user profiles.

3. Document Database search
The Document Database search Process allows user to perform retrospective searches against all items stored in the system.

key features
1. Document Database :-
- contains all items received and processed by the system
- Items are typically static and may be partitioned by time. for archiving proces

2. Queries :-
- users enter ad hoc queries to search the database

3. Search process
searches can span large time periods through users may constrain searches to recent items.

4. Index Database search (AFB)
The Index Database search process allows users to create and search indexes for efficient retrival of items

key aspects :-
1. Index files
users can create private Index files or use Public index files.
2. Automatic file Build
A process that automatically generates candidate index records for incoming items.

Relationship to Database Management Systems

IRS and DBMS serve different purposes but are increasingly being integrated for more efficient data management. The relationship stems from the need to handle both structured data and unstructured information

DBMS	IRS
structured (e.g., tables, records, attributes)	Unstructured or semi-structured (e.g document, articles, abstract)
well-defined schema with predefined relationships	free from text with minimal structure.
precise queries using SQL	keyword-based, full-text searches, relevance ranking.
exact matches, tabular format	Ranked Results, relevance-based.
Retrieves specific data Instantly	Iterative search process refining queries over time.

Integration of DBMS & IRS

Modern DBMS platforms incorporate IR features to handle both structured & unstructured data.

examples of Integration:-

- INQUIRE DBMS:
  One of the earliste systems to combine structured and unstructured data retrieval
- Oracle DBMS (connectis):
  Embeds an Information retrieval system that uses a thesaurus based search to improve accuraly.
- Informin DBMS (Retrieval ware):
  links structured databases with powerful search tools for text based data.

Importance of Integration

- Enables efficient data retrieval across structured and unstructured data sources.
- enhanies search capabilites with ranking, relevanle feedback, and thesaurs-based suggestions.
- supports business intelligence and decision making by providing a holistic view of data.

Digital Libraries and Data warehouses.

Digital libraries and Data warehouses (or Data Marts) are two major systems that overlap with IRS, but they serve distinct purposes.

while all three systems act as repositories of information, they differ in their focus, structure and retrieval mechanisms.

1. Digital libraries

- This originate from traditional libraries, which have always been concerned with storing and retrieving information
- As digital formats expanded, libraries evolved from electronic libraries (1991–1993) to digital libraries (post-1995)

challenges & opportunities in digital libraries

- Indexing & cataloging
- search Intermediaries
- Copyright & Intellectual property
- Digital preservation.

Digital libraries vs IRS

- similarities:
  Both store and retrieve textual Information
- Differences:
  - IRs mainly focuses on search & retrieval
  - Digital libraries address broader concerns like copyright, long-term accessibility & source authenticity.

2. Data Warehouses

- Originating in the commerical sector, data warehouses help organizations control, organize & retrieve structured digital data.
- Unlike digital libraries which focus on textual data, data ware houses handle structured, quantitative data for decision-making.

components:-

1. Data storage
2. Information Directory
3. Data processing & movement
4. Search & Retrieval tools
5. Data Export mechanism

Data warehouses vs IRS

- similarities
  Both allow users to search & retrive info
- Differences
  - Data warehouses deal with structured data for decision support & analytics
  - IRS focuses on textual, unstructured data with iterative searching and relevance ranking.

conclusion :- Integration of these systems while Digital libraries, Data warehouses and Information retrieval systems have distinct roles, they are increasingly integrated to provide comprehensive data management.

- Digital libraries enhance access to unstructured textual data.
- Data warehouses facilitate structured data analytics and decision making.
- IRS enables search, ranking and retrieval of relevant textual information.

Information Retrieval system capabilities

An Information Retrieval system (IRS) provides various capabilites that allow users to search, browse and interact with information effectively.

These capabilites are generally grouped into 3 categories.

1. Search capabilites

Search capabilites define how users query an IRS to find relevant information. A robust system offers multiple search options to accommodate different user needs and query types.

- Boolean logic

It has values True or False

Operator :- NOT, AND, OR

example :- 1. Mobile OR Laptop

Doc 1 (mobile, laptop) ✓
Doc 2 (mobile) ✓
Doc 3 (laptop) ✓
Doc 4 ✗

2. Mobile AND Laptop

Doc1 (mobile, laptop) ✓
Doc2 (mobile) ✗
Doc3 (laptop) ✗

- Proximity search

finds documents where 2 or more terms appear close together, within a specified distance.

e.g :- "data" Near/3 "mining" → finds "data" and "mining" within 3 words of each other.

- Contiguous word phrases

when 2 or more words that are treated as a single semantic unit.

Searches for exact seaquences of words in the same order.

e.g :- United state of America → only retrives document containing that exact phrases.

- Fuzzy logic search

- Allows approximate matching rather than exact matches
- Handles misspelling, typos or variations in text.

e.g :- Searching for "analyze" may also match "analyzes" or "analyze"

- Term masking (wild card search)

uses special characters to match partial terms or unknown characters.

e.g:- comput* → matches computer, computing, computation
test → matches test, tests, tent.

- Numerical and Date Range search.
  Finds documents that contain numbers or dates within a specified range.
  usecase:- especially useful in finance or academic database
  eg:- 2018 – 2022
  price [₹100 to ₹1000]

- Thesaurus Expansion (concept-Based search)
  Expands the search to include synonyms or related terms using a thesaurus or ontology.
  e.g:- search term "car" may also results containing "automobile", "vechicle".

- Natural language Queries.
  Allows users to types queries in everyday human language.
  e.g:- "what are the applications of artifical intelligence"?

- Multimedia Queries
  Enables search using non-textual input like images, audio, video or sketches.
  e.g:- Upload an image to search for visually similar images (e.g Google images)
  - Search for audio clips that sound similar.

Browse capabilities :-

It allow users to navigate through retrieved documents or databases without specifying precise search terms.
They are essential for exploratory search where users may not know exactly what they're looking for.

user —query→ IRS → Results → (Browse capabilities: serving results.)

- Navigate . • visualization • organize.

Key :-

1. Relevance ~~Capa~~ Ranking
   - Each retrieved item is assigned a relevance score (usually between 0.0 and 1.0)
   - Results are presented in ranked order, showing the most relevant items first.
   e.g:- A document with score 0.92 appears above one with 0.60.

user —query→ IRS —Result→ DOC 1, DOC2, DOC 3, DOC 4 ↓ Relvance Ranking
DOC 2
DOC 1
DOC 3
DOC 4.

2. Zoning :-

Only relevant parts (zones) of documents are shown
- Such as title, abstract, or specific matching passage.
- Helps users quickly assess relevance without reading the full document.

3. Highlighting

This capability significantly enhances the browsing and searching experience by making it easier for users to identify relevant information quickly.

- Matched terms are visually highlighted using bold text, color or underling.
- Helps users immediately spot why a document was retrived.
- Especially useful in long documents.

Browse capabilites enhance the user experience by helping users navigate, evaluate and refine search results more easily - especially when they're uncertain about their query or exploring a new topic.

Miscellaneous Capabilites.

These are additional features in an IRS that enhance, usability, flexibility and efficiency during search and retrival processes. They support both new and experienced users in refining and reusing the queiers.

1. Search History Log

It records all the queries submitted by the user during a session.

purpose:-
- Allows users to revisit previous searches
- Helps in refining or modifying past queries without retyping.
- useful for long research tasks iterative search

and comparative analysis.

e.g:- You search "machine learning", later refine to "machine learning in healthcare" - the system keeps both entries in the history.

2. Bookmarking

A bookmark saves the location or reference to a retrieved item, so the user can revisit it later.

purposes:- - Saves important or relevant documents
- used when users plan to return to the result after reviewing others.

eg:- In a digital library, you bookmark a journal article that you want to redd later.

3. Canned Query / stored Query

A canned query (also called a stored query) is a pre defined search query saved by the system or the user for future reuse.

purposes:- - Saves time for repetitive or common searches
- Can include placeholders or variables for runtime inputs.

e.g:-

A librarian creats a stored query for "title contains 'AI' AND year > 2020".

Each time it's used, the user just changes the keyword or year.

Miscellaneous capabilites improve the overall effectiveness and reusability of an IRS by enabling better interaction, reuse and system compatibility.

Unit - II

Cataloging and Indexing

- History and objectives of Indexing
- Indexing process
- Automatic Indexing

Information extraction Data structure

- Introduction to Data structure
- Stemming Algo
- Inverted file structure
- N-Gram Data structure
- PAT Data structure
- Signature file structure
- Hypertext and XML Data structure
- Hidden Markov models.

1. History of Indexing

-Ancient cataloging (before the 19th century)

- Babylonian libraries (3rd millennium BCE):
  Early forms of indexing appeared in cuneiform tablets, arranged by subject.

-medieval libraries:-
  Basic cataloging methods developed, but indexing remained mostly manual and localized.

19 & 20 century Developments.

- late 1800s:
  - The Dewey decimal system introduced hierarchial subject indexing.
  - libraries started organizing materials systematically.
- ~~19th & 20th century~~ Develo
- 1960s - MARC (machine Readable cataloging)
  -Developed by the library of congress (1966-68) for compelterized cataloging
  - Standardized bibliographic records for better sharing among libraries
- 1970's - Dialog system
  -first commerical indexing system, originally developed by Lockheed for NASA
  - Became a public service in 1978, doc expanding acess to scientific & technical

Modern Developments (1990s - present)

- 1990s :
  - full-text indexing became possible with digital storage and computing advances.
  - Indexing shifted from manual cataloging to automated systems using algo.
- 2000 - AI & ML
  - Semantic indexing and concept-based search started replacing keyword based indexing.
  - NLP improved search accuracy.

Objectives of Indexing

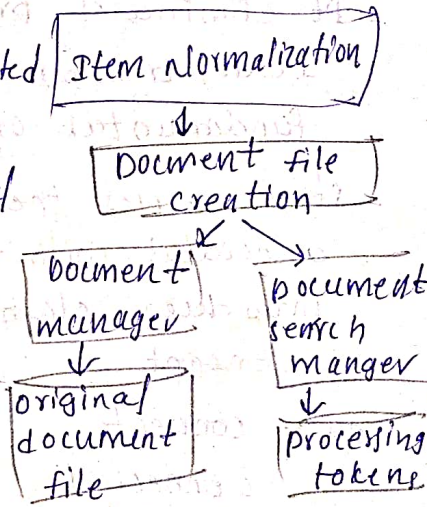
Indexing help users find relevant information efficiently. Its main objectives are:

- Enhance searchability & Retrieval
- Improve precision & Recall
- Support user needs
- Enable Ranking & clustering
- Adapt to Evolving Information Retrieval

Data structure

- each data structure has a set of capabilities
- Ability to represent the concept and their relationships.
- Two major data structures
  - stores and manages the recieved items in their normalized form (Document manager)
  - contains the preprocessing tokens & associated data to support search (Document search manager)

When searching is completed the result is passed to the document for retrieval



focus:- DS supports search fn

stemming :-

It is transformation often applied to data before placing it in searchable ds

→ It represents in canonical & morphological forms.

DS :-

1. stemming
   - porter stemming algo
   - Dictionary lookup stemmers
   - successors stemmers.

2. Inverted file structure
3. N-Gram Data structure
4. PAT Data structure
5. Signature file structure
7. Hypertext Data structure.

Stemming Algorithms

- Stemming Algo is used to improve the efficiency of IRS and improve recall
- It eliminates prefixes and suffixes from words transforming them into the their fundamental or root form.
- stem carries the meaning of the concept associated with the word and the ending introduce slightly modifications of the concept.

ex:- connect
connected
connecting
connection

} The major use of stemming is to improve recall.

Stemming Algos

- Table lookup stemming
- porter stemming
- Dictionary stemming
- successor stemming

Table lookup

- It store a table of all index terms and their stems, so terms from queries and indexes could be stemmed very fast.
- Its constructed with stem for every word
- systems use a combination of dictionary lookup and conflation rules

problems :-

There is no such data for English.

porter stemming

- A stem is a part of word
- In porter stemming following some conditions based on vowels and consonants.

ex:- connecting have 'ing' suffix to eliminate the suffix we can perform stemming

- This algo used to remove the suffixes from an English word and obtain its stem.

eg:- TREE
C C V V
V V
C V

[C] [VC]^m [V] → no. of VC(s)
m=0

TROUBLE
C C V V C C V
V V V I
C V C V

[C] [VC]^m [V]

eg:- 'sses' → replace with 'ss'
stresses → stress
'Ing' → nothing
making → make

'At' → Ate

Inflated — Inflate

'y' — i

happy → happi

'Able' → Nothing

Adjustable → Adjust

Dictionary lookup stemming

- Tries to avoid collapsing words with different meaning to same root
- the original word or stemmed version is looked up in a dictionary and replaced by the best stem.
- This technique has been implemented in Inquery and Retrieval ware system..

Inquery system technique called kstem.

- kstem is a morphological analyzer that conflates words variants to a root form.
- It uses 6 major data files to control & limit the stemming process.

Successor stemmers.

→ It is based on the length of prefixes

- The smallest unit of speech that distinguish on word from another
- these process uses successor varieties for a word.
- There once the algorithm takes substrings of a term will decrease as more characters are added until a segment boundary is reached.
- The stemming method based on this work uses letters in place of phonemically transcribed, utterances.

we can use 4 methods to successor varity of words

cutoff method:-

A cutof value is selected to define the stem length.

segment when successor variety >= threshold

Ex:- READABLE

corpus:- ABLE, APE, BEATABLE, FIXABLE, READ, READABLE, READING, READS, RED, ROPE, RIPE

prefix	Successor	letters
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,E,S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	Blank

Peak and plateau:-

- A segment breaks is made after a character whose successor variety exceeds that of the character.
- break at the charaiter whose succesor variety is greater than both its preceding and following charactw

READ | ABLE

complete word method

Breaks on boundaries of complete words

- Its made if the segment is a complete word in the corpus (READ)

Entropy methods-

uses the distribution method of successor varity letters.

$$\frac{|D_{ij}|}{|D_i|}$$

$|D_i|$ - The no. of words in a text body beginning with the i length of letter a

$|D_{ij}|$: The no. of words in D_i with the sucessor j

Inverted file structure

Vinay kumar

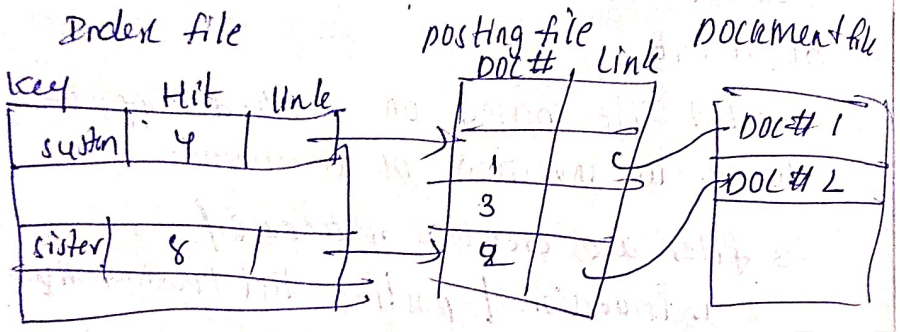
- It is a data structure, it allows efficient, full text searches in the database.
- It stores a mapping of words to their locations in the database table or document.
- Inverted file based on methodology of storing an inversion of document

3 files are created or accesed

1. Inversion / posting list / posting file
2. dictonary / Index file
3. Inverted file

- For each word a list of documents in which the each word is found is stored.
- Each document is given a unique the numerical identifier that is stored in inversion list which is stored list in the system and a pointer to the location of the its inversion list.
- using zoning to improve
- Inversion list consists of document identifier for each document in which the word is found.

- Inversion list are used to store concept and their relationship.
- words with special characteristics can be stored in their own diretory



N- Gram Data structure

- N Grams can be viewed as special technique for conflation and as a unique data structure in Ds
- These are fixed length of consecutive series of 'n' characters
- In stemming we care about stemantic but n gram we don't care.
- Here we are observing
  - →unigram- how often a word occur without looking at previous word
  - →Bigram - only the previous word to predict the current word
  - → Trigram - previous two words are considered

Here using # is to represent the interword symbol which anyone of a set of symbol.

- Each of the n-grams created becomes a separate processing tokens and searchable.
- It is possible that the some n-grams can be created multiple times from a single word.

$$P = (wi / wi-1)$$

$$\rightarrow \frac{count\ (wi-1, wi)}{count(wi-1)}$$

1 gram — no word history
2 gram — 1-gram word history
3 gram — 2-word history.

Hello! where are you ...
now?
from?
going?

Indexing process and cataloging

Indexing:

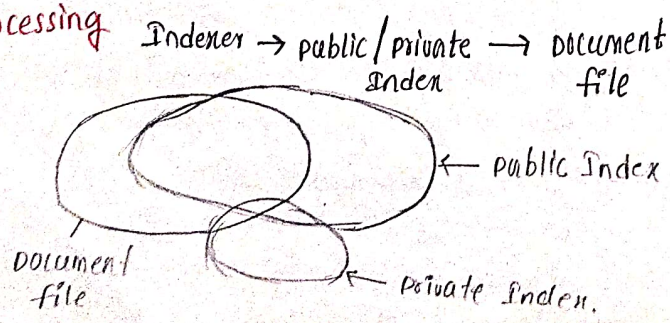
The transformation from received item to searchable data structure is called indexing.

- Indexing process can be manual or automatic
- Creating a direct search in document database or indirect search through index files.
- Once the searchable data structure has been created, techniques must be defined the correlate the user entered query statement to the set of items in DB.
- Indexing originally called cataloguing.

Objectives:-

- Represent the concepts within an item to facilitate the users finding relevant information.
- The full text searchable data structures for items in document file provides a new class of indexing called total document indexing.
- Other objectives are indexing are ranking, item clustering.

Index processing



When an organization with multiple indexers decide to create a public or private index.

- Scope of the indexing - define what level of detail the subject index will contain.
- Second decide to link index term together in a single index for a particular concept.

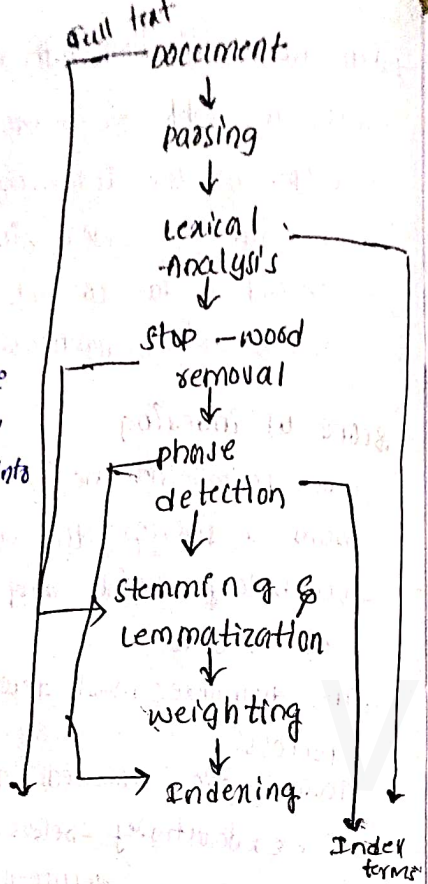
Scope of indexing

- when performing the indexing manually, problems arise from 2 sources the author and the indexes
- vocabulary domain may be different the author and the indexes.
- The indexer must determine when to stop the indexing process.
- Two factors involved in deciding
  - exhaustivity - refers to how many concepts from a document are indexed.
  - Specificity:- How precise or general the index terms are.

Precoordination and linkages

- Precoordination is a process of linking related index related index terms during the indexing phase of document processing.
- Post coordination:- which index terms are not linked during indexing, and the relationships are established during search time, its called post coordination.
- Use of linkages:-
  - linkages connect attributes of concept (e.g., actor, action, location).
  - Help disambiguate similar terms

Text processing



1. Document Parsing:-

Documents come in all sorts of the same document may contain multiple languages or formats.

- Document parsing deals with the recognition and "breaking down" of the document structure into individual components.

2. Lexical Analysis

Document parsing after lexical analysis tokenizes a document, seen as input stream into words.

3. Stop-word Removal:-

A subsequent step optionally applied to the results of lexical analysis is stop-word removal - the removal of high-frequency words.

4. Phrase detection

This step captures text meaning beyond with pure bag-of-word approaches, thanks to the identification of noun groups and other groups.

5. Stemming and lemmatization:

stemming & lemmatization aim at stripping down word suffixes in order to normalize the word.

- stemming is a heuristic process that "chops off" the ends of words in the hope of achieving the goal correctly most of the time.
- lemmatization is a process that typically uses dictionaries and morphological analysis of words in order to return the base or dictionary form of a word, thereby collapsing its inflectional forms

Automatic Indexing :-

It is the process by which a computer system automatically selects and assigns index terms to a document, without human intervention.

It is faster, more scalable and less biased compared to manual indexing.

- Automatic indexing having advanced technologies such as natural language processing (NLP) and machine learning.
- It finds relevant terms or keywords from the documents automatically and create an index.
- first system scans the text and identify important words to create an index

How it works?

1. Document parsing
2. Text preprocessing
3. Term selection.
4. Term weighting
5. Index creation.

Advantages:-

1. speed
2. consistency
3. scalability
4. cost-effective.

Challenges:-

- may miss contextual meaning
- Ambiguity & polysemy
- Harder to handle domain-specific language without proper training.

Inverted file structure

It is mostly widely used data structure in Information Retrieval systems. It is used to map terms (words) to the list of documents in which they appear. This is the core of fast full-text searching.

Why Inverted?

Because instead of listing all words per document, we store all documents per word - which is the inverse of the document structure.

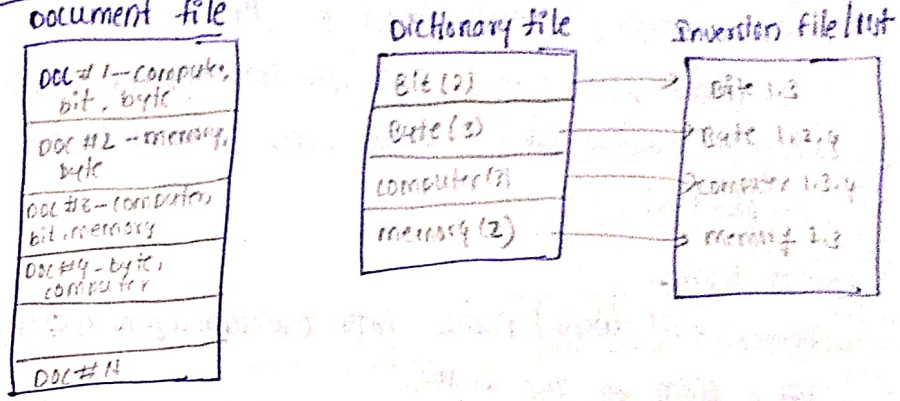
components

1. Document file : stores the actual documents
2. Dictionary : Stores all unique terms in the document collection in sorted order.
3. Inversion list / posting list :-
   for each term in the dictionary, stores a list of document IDs where the term appears.

How it works:-

1. Each document is assigned a unique numeric Document ID.
2. For every term an inversion list (posting list) stores:
   - The document IDs where the term occurs.
   - Optionally : positions, frequencies or weights.

ex:-



Additional features

- Zoning: limits search to certain zones to increase precision
- weights
- Special Dictionaries
- Ranking support.

Advantage:-

- Fast search
- scalable
- flexiable
- supports Ranking.

The inverted file structure is the backbone of modern Information Retrival. It allows fast, flexible and precise searching by mapping terms to documents, enabling efficient query processing even at massive scale.

N-Gram data structure :-

A N-Gram is a sequence of n consecutive characters extracted from text. It used to analyze, search and match parts of words or phrases without considering the meaning (semantics).

Why N-Gram:-
- Unlike stemming, which looks for the root of a word
- N-Grams break text into overlapping fixed-length tokens, helping match terms even if misspelled or partial.

How it works:-
1. Convert each word / phrase into overlapping N-grams.
2. Store them in the Index.
3. During search, the query converted into N-grams and matched.

$$P(w_i / w_{i-2}, w_{i-1}) \rightarrow \frac{count(w_{i-2}, w_{i-1}, w_i)}{count(w_{i-2}, w_{i-1})}$$

Statement:-

DO I like ....

corpus DataBase
I am Amit
I like computer
DO Amit like computer
Do I like computer
I do like Amit.

DO ....

Next word	Frequency	probability
I	6	2/4
Am	2	0/4
Amit	5	1/4
like	6	1/4
computer	5	0/4
do	4	0/4

1. $w_i = I$, $w_{i-1} = do$ → $\frac{count(do\ I)}{count(do)} = \frac{2}{4}$

2. $w_i = am$, $w_{i-1} = do$ → $\frac{count(am, I)}{count} = \frac{0}{4}$

3. $w_i = like$, $w_{i-1} = do$ → $\frac{count(do\ like)}{count(do)} = \frac{1}{4}$

4. $w_i = Amit$, $w_{i-1} = do$ → $\frac{count(Amit, do, Amit)}{count(do)} = \frac{1}{4}$

⇒ DO I like, in this way calculated.

PAT Data Structure

Practical Algorithm to Retrieve Information coded in Alphanumeric

- A PAT Tree (or PAT Array) is a compressed binary tree used to efficiently store and search substring of text.
It is especially useful in string matching, prefix searching and text indexing.

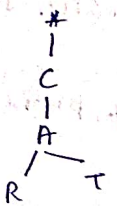
Key characteristic :-
- Designed for continuous text input
- Works with unique substrings called sistrings
- A binary tree structure that supports logical sorting
- Supports range searches better than inverted file
- each positions in the input becomes an anchor point for a sistring.
- A sistring is a substring starting from a specific character position to the end of the input text.

How it works:-
1. Every position in the input string becomes a starting point for a sistring
2. All sistrings are sorted as leaf nodes in a binary tree.

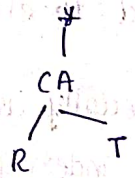
3. Intermediate nodes store bit positions where strings differ.
4. The comparison is done bitwise for efficiency.
5. The tree is compressed to reduce size by skipping redundant bits.

e.g:- CAT → C A T = 3 bytes
1 byte 1 byte 1 byte
CAR → C A R = 3
6 bytes.

Regular tree



Compact PAT tree

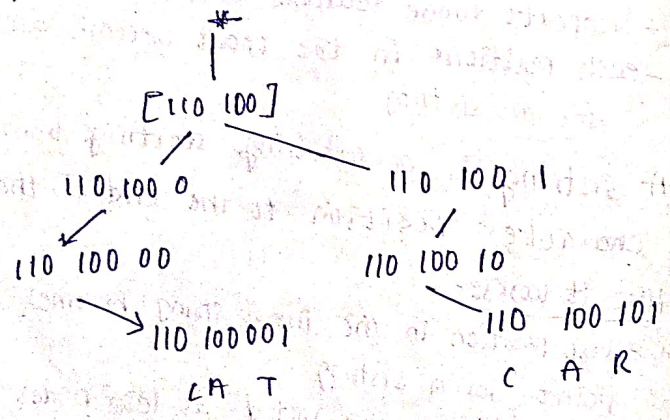


Binary Representation

	C	A	T
CAT =	110	100	001
CAR =	110	100	101
	C	A	R

C = 110
A = 100
R = 101
T = 001

0 - left
1 - Right



Advantages:-
1. Efficient prefix search
2. Compact Representation
3. logical sorting
4. Better than Inverted file

Disadvantages:-
- complex to implement and maintain
- not ideal for datasets with few repeated patterns
- Requires binary conversion for substring processing.

Signature file structure.

A signature file is a filtering-based data structure used in Information Retrival to quickly eliminate non-matching documents before doing a full-text check. It uses bitwise codes (signatures) to represent the presence of terms in documents.

Key points:-
- Each word is converted into a fixed-length binary code (word signature) using a hash function.
- The bit positions set to 1 in the signature depend on the hash output.
- All word signatures in a document are OR-ed together to form a block / document signature.
- During search, a query signature is generated and matched against document signatures using bitwise comparision.

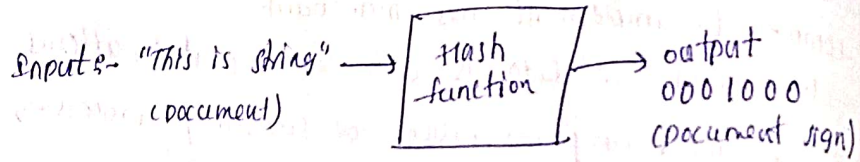
creation of signatures

↓

Hash function is used for creating signatures.

↓

Hashing is a process that transforms input data of any size into a fixed-size value or key, typically a string of characters or a numerical value.



Ex:- "computer science graduate students study"

word	signature
computer	→ 0001 0110 0000 0110
science	→ 1001 0000 1110 0000
graduate	→ 1000 0101 0100 0010
students	→ 0000 0111 1000 0100
study	→ 0000 0110 0110 0100

ORing

	0110
	0100
	0110

↓

ORing

signature → 1001 0111 1110 0110

user → query computer: ~~1001 0111 1110 0110~~ 001 0110 0000 0110 → 1001 0111 1110 0110 (searching of matches)

1001 0111 1110 0110

AND. 0001 0110 0000 0110

0001 0110 0000 0110 → match found

Advantages:-

- compact structure, - fast filtering, - parallel processing
- works with WORM media
- low-frequency terms.

Limitations:-

- false positives: Because unrelated documents may match query signature bits
- No ranking: Unlike inverted indexes, it doesn't easily support term weighting or ranking.
- less efficient for very large or frequently updated collections.

Hidden markov model

Hidden markov model (HMM) is a statistical model used to represent systems that are probabilistic and sequential where the system being modeled is not directly observable (hidden), but can be inferred through observable outputs.

HMMs have been sucessfully applied in:

- speech recognization
- Named Entity Recognition (NER)
- Optical character recognition (OCR)
- Topic identification.
- Information Retrival.

concept :-

A markov model assumes:

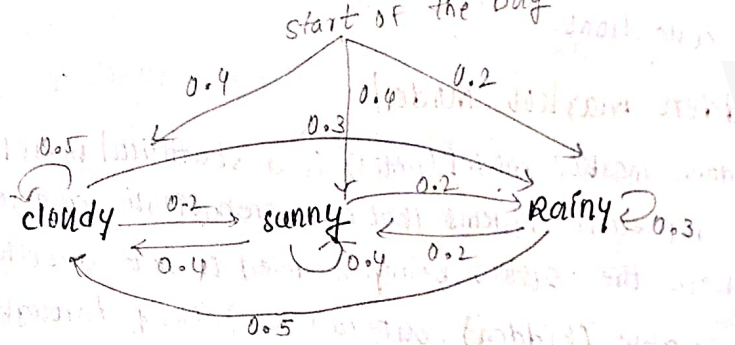
- The system is in one of a finite no. of states.
- The next state depends only on current state

(not full history) - called a first order markov property.

How it works:-

1. The model starts in an initial state.
2. It transitions to the next state using transition probabilites (A).
3. At each state, it emits an o/p symbol using output probabilites (B).
4. The actual sequence of states is hidden, only outputs are seen.

Ex:-



Present Day \ Future Day →	cloudy	sunny	Rainy
cloudy	0.5	0.2	0.3
sunny	0.4	0.4	0.2
Rainy	0.3	0.2	0.5

initial

cloudy	sunny	Rainy
0.4	0.4	0.2

cloudy (event 1) → sunny (event 2) → cloudy (event 3) → Rainy (event 4)

$P(E_2, E_3, E_4 | E_1)$

↓

P(init Prob. of E_1) * $P(E_2|E_1)$ * $P(E_3|E_2)$ * $P(E_4|E_3)$

probability = 0.2 * 0.4 * 0.3 * 0.4

Event 1	Event 2	Event 3	Event 4	
(cloudy / sunny / Rainy)	(cloudy / sunny / Rainy)	(cloudy / sunny / Rainy)	(cloudy / sunny / Rainy)	
3	* 3	* 3	* 3	= 81

1. cloudy → sunny → Rainy → cloudy = P1
2. cloudy → Rainy → Cloudy → sunny = P2
⋮
81. sunny → cloudy → Rainy → Rainy = P81

Applications:-
- Ranking documents
- sequence modeling
- Language modeling.

Advantages:-
- Handles uncertainity & sequential patterns
- can model temporal or structured data
- useful for ranking, prediction and patter detection

Limitations:-
- computationally intensive for large state spaces
- Requires training data
- Assumes Markov property.

Hypertext and XML data structures in Information Retrieval

What is Hypertext?

It is a form of text that enable includes links (hyperlinks) connecting it to other text, documents, or multimedia resources.

This structure enables non-linear navigation and is the foundation of the world wide web.

key:-
- supports inter document and intra-document linking
- Helps user browse and navigate through information intuitively.
- Typically represented using HTML.

In Information Retrieval (IR)
- Hypertext adds context and connectivity to documents
- used in web search engines, digital libraries, and hypermedia systems.
- links help in Ranking.

What is XML? (Extensible markup language)

xml is a structured document format that defines rules for encoding documents in a way that is both human-readable and machine readable.

structure:- XML documents are made of
- Elements (tags like <title>, <author>)
- Attributes
- Hierarchical tree structure.

Why use XML in IR?
- precise tagging and structuring of information
- storing semantic relationships blw parts of a document
- Efficient querying using XML-aware search engines (e.g. xpath, xQuery).

-Applications:-
- Digital libraries
- web search
- Enterprise document management
- Semantic web & RDF search engines.

conclusion:-
- Hypertext helps organize documents non-linearly, enabling web-style navigation.
- XML provides structure, semantics and metadata to documents, making them easier to index, search and retrieve.
- Together, they enrich document representation & retrieval in ~~Random~~ ~~IR~~ modern IR systems.

Unit - III

Automatic Indexing

- classes of Automatic Indexing
- Statistical Indexing
- Natural language
- concept Indexing
- Hypertext linkages

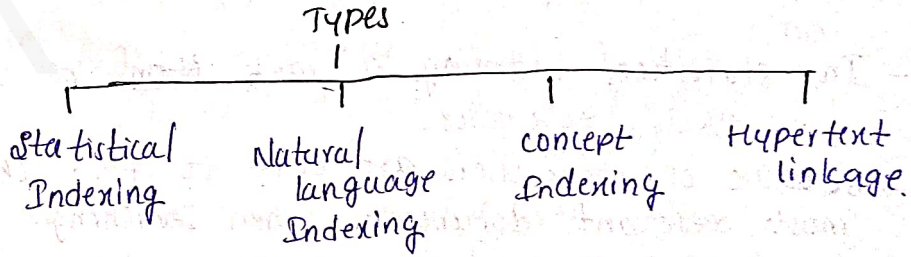
Document and Term clustering

- Introduction to clustering
- Thesaurus Generation
- Item clustering
- Hierarchy of clusters.

Automatic Indexing

- A method of indexing in which an algorithm is applied by a computer to the title/text of a work to identify and extract words and phrases representing subjects based on entries of index

- Automatic indexing is the process of assigning keywords or index terms to documents without human intervention, using computational techniques. It helps in organizing and retrieving information efficiently.



- It requires few seconds based on the processor & complexity of algo to generate Indexes
- It is the process of analyzing an item to extract the information to be permanently kept in index.
- Here we can associate with generation of the searchable data structure associated within an item.

→ Automated indexing fall into two classes
- weighted
- unweighted

Statistical Indexing

Statistical Indexing is a technique used in information retrieval to rank and retrieve documents based on statistical properties of terms in collection.

Instead of relying on predefined keywords it analyzes term frequency and distribution patterns to improve search accuracy.

(or)

- In statistical indexing is rank items based on their relevance.
- Because of their users are easy to find the most relevant documents when searching.
- The documents are found by normal boolean search and then statistical calculation are performed on the hit file.
- Ranking the output
- Statistical indexing based on upon different models. Those are

1. probability weighting:-
   Assigns weights based on the probability of a term appearing in relevant vs non relevant documents.
2. vector weighting:-
   Represents documents as vectors and ranks them using cosine similarity.
3. Simple Term frequency Algo:-
   Weighs terms based on how often they appear in a document.
   a. Frequency of occurance
      To calculate a relevance value for each item. The frequency is used to determine the importance of an item in relation
   b. Document frequency:-
      It refers to the no. of documents in a corpus that contain a particular term.
   c. Term frequency
      It refers to the no. of times a particular term appears in a document.
4. Inverse Document frequency Algo
   Gives higher weight to rare terms
5. Signal weighting:-
   Uses signal processing techniques to detect important terms.
6. Discrimination weighting
   Assigns higher weight to terms that differentiate relevant from non-relevant documents.
7. Bayesian model:-
   Uses Bayes Theorem to calculate the probability of a document being relevant

These weighting techniques are used to improve information retrieval, search engine ranking, document classification.

Natural language

The goal of natural language processing is to use the semantic information in addition to the statistical information to enhance the indexing of the item.

- It improves the precession of searches reducing the no. of false hits a user receives.
- The information is extracted as a result of processing the language rather than treating each word as independent entity.
- After the processing the output is generation of phrases that become index to an item.
- more complex analysis generates thematic representation
- word phrases generated by natural language processing algorithms. index specification and provide another level of disambiguation.

In Natural language

1. Index phrase Generation
2. Natural language processing.

Index phrase Generation:-

The goal of indexing is to represent the semantic concepts of an item in the information system to support finding relevant information.

- single words have conceptual context, these are very easy to find normal info.
- Term phrases allow additional specification and focusing of concept to provide better result except non-relevant items.

Natural language processing

It provides higher-level semantic information identifying relationships btw concepts.

1. The processing tokens in the document are mapped to subject codes.

↓

2. Text structure, which attempts to identify general discourse level areas within an item.

↓

Assignments of terms to components, classifying the intent of the terms in the text.

↓

Identifies interrelationships btw the concepts.

↓

finally assign final weights to the established relationship.

Advantages :-

- Better precision
- phrase normalization
- Disambiguation
- Context-aware
- Improves ranking.

Challenges :-

- language is ambiguous and complex
- Requires large knowledge bases & training data
- computationally intensive compared to keyword matching.

Concept Indexing

In previous indexing i.e Natural language processing starts with a basis of the terms and extends the information to phrases.

- Higher level concepts builds relationship blw concepts
- concept indexing takes the abstraction a level further.
- its starts with a no. of unlabeled concepts:

e.g: automobile - term.

concepts are- vechicle, Transportation, Environment. fuel, mechanical devices.

vechicle - 0.65
Transportation - 0.60
Environment - 0.35
fuel - .33
Mechanical - .15.

Concept Indexing means indexing a document based on the meaning of words, not just the exact words.

- It focuses on what the document is talking about, not just which words are used.

Why do we need it

Sometimes, different words mean the same thing

for e.g

- car = automobile
- "Heart attck" = myocardial infarction
- mobile = "cell phone"

If you search for "automobile, but the document says "car", a basic system may miss it. But concept indexing will understand both mean the same and show the result.

How it works

concept indexing uses tools like

- Thesaures (like a smart dictionary of related words)
- Ontologies (big knowledge maps of terms & meanings)

These tools help group words with similar meanings into the same concept.

Advantages :-

- finds similar meaning
- Handles synonyms
- Better search accuracy
- Good for smart systems

Drawbacks

- Needs smart tools
- slower than simple search
- Errors possible.

Hyper Linkages (Hypertext linkages)

Hypertext linkages are clickable links that connect one document to another.

In IRs, hypertext linkages help in

- connecting related documents
- providing extra information
- And potentially improving search results by including linked content.

Why they are important

- Traditional documents are 2D - they only contain text.
- with hypertext, documents become 3D - they contain text connections to other documents.

How can Hypertext linkages be used in Indexing?

Basic Idea:
Treat the content in linked documents as part of the main document - but with less weight.

> That means: if Document A links to Document B then some keywords from B should be added to A's index, but with reduced importance.

weight formula
- Direct text = high weight
- Linked content = lower weight
- The closer the link is to relevant text, the stronger its influence.

There are 3 main ways hypertext is currently used

1. Manual Indexes:
   e.g: YAHOO
   - People organize websites into categories.

2. Automatic Indexes:-
   e.g:- bYCOS, Altavista
   - Bots read pages and index words automatically

3. Web Crawlers:-
   e.g:- Google, webcrawler
   - Bots follow hyperlinks across websites to build index.

Advantages:-
- connects related info
- Better search results
- Helps navigation

Challenges:-
- link passing errors
- synonym confusion
- overhead
- link spam.

Document and Term clustering

Introduction to clustering

clustering is the process of grouping similar items or terms together into categories called clusters (or class) (or classes)

In information Retrival (IR), clustering helps:-
- Group documents discussing the same topic
- Organize terms that have related meanings (thesaurus)
- Improve search effectiveness by finding related results.

clustering useful for organizes data, Improves recall, enhances searching.

2 main uses of clustering

1. Term clustering (Thesaurus Generation)
   Groups similar terms, like synonyms
   - Improves query expansion.

2. Document clustering
   Groups similar documents, even if exact query terms don't match.
   - Improves retrieval of similar content.

Steps in clustering proces

1. Define the Domain
   - what are we clustering (e.g medical terms, articles, emails)

2. Determine attributes
   - choose what part of the document to consider (title, abstract, body)

3. measure Relationship.
   - use co-occerence of words or similarity functions (like cosine similarity.

4. Apply clustering Algorithm
   - used a method like k-Means or hierarchical clustering to form groups

Challenges in clustering:-

- Ambiguity of language
- Too much recall can reduce precision
- Homographs (same word, different meanings)

Thesaurus Generation

A thesaurus in information (IR) is collection of related terms (like synonyms hierarchical or associated terms) used to improve search precision and recall by expanding queries.

A book or electronic resource that list words in groups of synonyms and related concepts.

Types of Thesaurus Generation

1. Manual clustering

- Human experts manually group words into conceptually similar clusters.
- Relies on knowledge of:
  - Domain (e.g medical, legal)
  - Concordances (word frequencies + occurances)
  - Tools like
    - KWOC (Keyword out of context)
    - KWIC (Keyword in context)
    - KWAC (Keyword and context)

pros:- High precision cons:- Time consuming, hard to scale.

Item: An item could be document, a web page, an image, a video, or any other piece of information that can be indexed and searched for

Term:- A "term" refers to a significant word

2. Automatic term clustering

Uses statistical analysis to group similar terms

key idea:- words that appear togther in many documents likely belong to the same concept.

Techniques for Automatic thesaurus creation:-

1. Complete Term Relation method / Term clustring
   - compares every pair of words using similarity measures (like co-occurrence).
   - creates a Term-Term matrix (word vs word)
   - Thresholding: If similarity > threshold, they're grouped.

clustering algorithms used:-
- Cliques: All terms are similar to each other (high precision)
- Single link: If one term matches, add it (high recall, low precision)
- star: One core term, add all related terms
- string: like a chain of connected terms.

Term clustering in information retrival is the process of grouping similar terms together based on their occurrence and context within the dataset.

e.g:- Given Items / Documents & their extracted terms:-

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8
Item 1	0	4	0	0	0	2	1	3
Item 2	3	1	4	3	1	2	0	1
Item 3	3	0	0	0	3	0	3	0
Item 4	0	1	0	3	0	0	2	0
Item 5	2	2	2	3	1	4	0	2

Formula= $SIM(Term_i, Term_j) = \Sigma (Term_{k,i}) (Term_{k,j})$

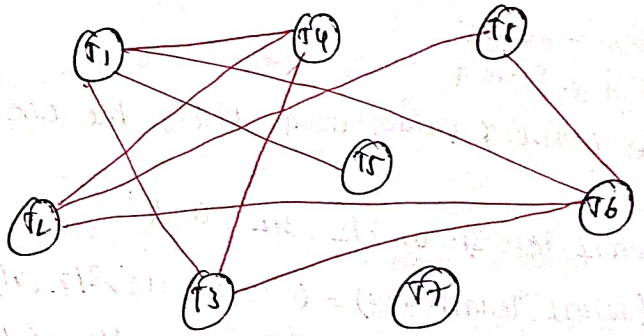
	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7	Term 8
Term 1		7	16	15	14	14	9	7
Term 2	7		8	12	3	18	6	17
Term 3	16	8		18	6	16	0	8
Term 4	15	12	18		6	18	6	9
Term 5	14	3	6	6		6	9	3
Term 6	14	18	16	18	6		2	16
Term 7	9	6	0	6	9	2		3
Term 8	7	17	8	9	3	16	3	

The next step is to select a threshold that determines if 2 terms are ~~associated~~ considered similar enough to each other to be in the same class. Threshold value ~~is~~ of 10 is used.

Let take Term as T

	T1	T2	T3	T4	T5	T6	T7	T8
T1		0	1	1	1	1	0	0
T2	0		0	1	0	1	0	1
T3	1	0		1	0	1	0	0
T4	1	1	1		0	1	0	0
T5	1	0	0	0		0	0	0
T6	1	1	1	1	0		0	1
T7	0	0	0	0	0	0		0
T8	0	1	0	0	0	1	0	

cluster using single link



ii. Clustering using Existing Clusters

- start with predefined clusters
- compute centroids (average position in vector space)
- Re assign terms to clusters based on similarity to centroids
- Repeat until stable (iterative refinement.

- pros : fewer calculations
- cons :- Need to predefine no. of clusters.

iii. One pass assignments.

- Very fast method (O(n))
- first term forms a new class
- compare each new term to existing class centroids
- If similar → adds to class; if not → create new class

• Pros: Efficient
• Cons: Result depends on input order, not optimal.

e.g :- from table t2 in previous page.

class 1 = Term 1, Term 3, Term 4
class 2 = Term 2, Term 6, Term 8
class 3 = Term 5
class 4 = Term 7

Note the centroid values used during the one-pass process :-

class 1 (Term 1, Term 3) = 0, 7/2, 3/2, 0, 4/2
class 2 (Term 1, Term 3 Term 4) = 0, 10/3, 3/3, 3/3, 7/3
class 1 (Term 2, Term 6) = 6/2, 3/2, 0/2, 1/2, 6/2.

Advantages

- Improved Recall
- Better precision
- query expansion
- semantic search
- Supports NLP & AI

Drawbacks

- Time consuming (manual)
- Limited domain generalization
- Ambiguity & Homographs
- Over generalization

Applications :-

* IRS
* Knowledge Representation
* Search Engines
* Digital libraries.

Item clustering (Document clustering)

Item clustering is the grouping of documents / items based on their content similarity. The main goal is to cluster togther items that talk about similar topics so that users can:

- find related documents easily.
- Expand or refine search results

How it works :-

Item clustering uses similarity functions - just like term clustering - but instead of comparing terms across documents, we now compare entire documents. using their term vectors.

The similarity blw Item calculated using a vector model, and various clustering algorithms are applied such as :

- Clique method
- single link method
- star method
- string method.

examples :-

Document	content
DOC 1	– "cats are lovely animals that love to sleep".
DOC 2	– "Dogs are loyal and active pets"
DOC 3	– "Cats and dogs can live together peacefully.
DOC 4	– "python is popular programming language."
DOC 5	– "python & Java are used in software development".

Step 1 :- convert document to term vectors

Term	DOC 1	DOC 2	DOC 3	DOC 4	DOC 5
cats	1	0	1	0	0
dogs	0	1	1	0	0
python	0	0	0	1	1
Java	0	0	0	0	1
programming	0	0	0	1	0

Step 2 :- Measure similarity

We compare documents based on shared terms.

- DOC 1 and DOC 3 share "cats"
- DOC 2 and DOC 3 share "dogs"
- DOC 4 and DOC 5 share "python"

Step 3 :- cluster them

Using a simple single-link clustering, we get.

cluster	documents.
cluster A	- DOC 1, DOC 2, DOC 3 → all about pets (animals)
cluster B	- DOC 4, DOC 5 → both about programming

Results :-

Now if a user searches for "cats", the system may also show DOC 3 (because it's in the same cluster) even if "cats" isn't the top keyword → this improves recall

- If the user searches "Java", they'll see DOC 5 and also DOC 4 because they are clustered together, helping the user discover related info.

Benefits :-

- Enhance navigation in large datasets
- Improves search relevance
- Enables query expansion

Drawbacks

- precision loss
- Topic mixing
- Computational cost.

Hierarchical clustering

Hierarchical clustering is a method used to group documents or terms into a tree-like structure of clusters. It shows which documents are closely related, which are broader and how they are nested within each other.

Applications :-

- used in systems like Scatter/Gather
- Helps in browsing, filtering & query expansion
- used in Yahoo style - category systems.

Challenges & Drawbacks :-

- Automatic hierarchy creation (especially for terms) is error-prone due to language ambiguity
- If poorly done, can reduce recall or precision
- manual hierarchies are better for users but require effort.

(b) term vectors

1. Explain briefly about hierachy of clusters

- It builds hierarchy of clusters.
- This algo starts with all the data points assigned to a cluster of their own.

The two nearest clusters are merged into the same cluster.

- A technique to create a tree like structure (hierarchy) of clusters from documents or terms based on similarity.

Two main types:-

1. Agglomerative (Bottom-up):- start with individual items, merge similar ones.
2. Divisive (Top Down): start with one big cluster, split into smaller ones.

* Hierarchical Agglomerative clustering (HACM) is widely used in IRS.

Goals or objectives of Hierarchical clustering

1. Improve search efficiency
2. Enhancing user experience
3. Enhancing Data organization
4. Reduce information overload
5. Improving Relevance of search results.

Key concepts:-

- Dendrograms: Tree-like diagrams that show the merging or splitting of clusters.
- Centroids: Represent the average vector of items in a cluster.
- Ward's Method :-
  uses minimum variance and Euclidean distance to form compact clusters.

$$I_{ij} = ((m_i m_j) / (m_i + m_j)) d_{ij}^2$$

$$d_{ij}^2 = \Sigma_{k=i} (x_{ik} - x_{j,k})^2$$

- Lance-Williams formula:-
  General formula for calculating dissimilarity between clusters.

$$D(C_{ij}, C_k) = \alpha_i D(C_i, C_k) + \alpha_j D(C_j, C_k) + B D(C_i, C_j) + \gamma |D(C_i, C_k) - D(C_j, C_k)|$$

Cluster Representation

- Monolithic clusters: focused on a single topic
- Polythetic clusters:
  Represented using top keywords from multiple ~~sub~~topics.

Hierarchial clustering provides a structured and scalable approach to organizing large documents collections, supporting both efficient search and intuitive browsing.

Unit-IV

User search Techniques:-

- Search statements, and Binding
- Similarity measures and Ranking
- Relevance feedback
- Selective Dissemination of Information search.
- weighted searches of Boolean systems
- Searching the Internet and Hypertext.

Information Visualization:

- Introduction to Information Visualization.
- cognition and perception.
- Information Visualization Technologies.

Search statement are the statements of an information need generated by user to specify the concepts they are trying to locate in items.

User search techniques are

1. Search statements
2. Binding (Query Binding.

Search statement and Binding

Search statement (Query):

- A search statement is essentially the query that a user inputs into an information retrieval system.
- This can include keywords, phrases or even more complex queries involving Boolean operators (like AND, OR, NOT)
- The primary purpose of search statement is to define the user's information need as precisely as possible.

ex:-

Basic search statement:-

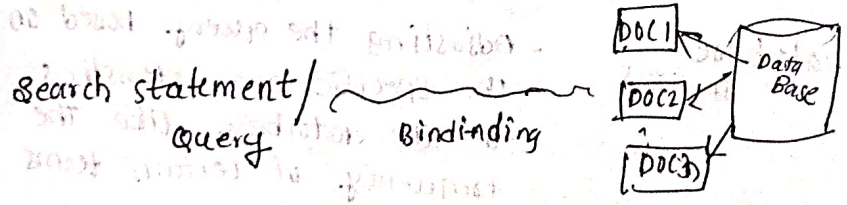
"Climate change and Arctic wildlife".

Advanced:-

"Climate change AND Arctic wildlife NOT polar bears

Binding (or) Search statement Binding (or) Query Binding

Binding refers to the process of matching the search statement or query with the relevant data within the database.



- Binding is to the vocabulary and past experience of the user.
- The search statement is the users attempt to specify the conditions needed to logically subset the total items space to cluster of items that contains that information needed by the user.

Binding (first level):- (user specific)
- How the users own knowledge and vocabulary shape their search query
- When user create a search statement they are using their understanding and past experience to form a query that captures what they need from database.

Binding (second level) (system specific)
- The next binding level happens when the search system processes the users query
- It translates the query into a standard format that the system can understand (Query processing)

Binding (third/final level) (database specific)
- Finally, the search query is applied to a specific database
- Adjusting the query based on the specific characteristics of the database, like the frequency of certain terms.

example :-

	Input	Binding	
i)	"find me information on the impact of the oil spills the Alaska on the price of oil"	user search statement using vocabulary of users.	Level-1
	impact oil (petroleum), spills (accidents), Alaska price (cost, value)	statistical system binding extracts processing tokens.	Level-2
	impact (.308), oil (.606), petroleum (.65), spills (.12) accidents (.123) Alaska (.45) price (.16) cost (.25), value (.10)	weights assigned to search terms based upon inverse document freqency algo & database	Level-3

Similarity Measures

- Similarity measures is a function which is used to measure the similarity blw user query and documents.
- It is possible to retrieve documents in the order of presumed importance.
- This function calculates the degree of similarity blw a pair of text objects.

Similarity measure methods

1. Cosine similarity method
2. Jaccards method
3. Dice method.

Cosine similarity method

It is a measure used in information retrieval and text mining to determine how similar two documents (or a query & a document) are in terms of their content.

$$\text{Similarity}(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

cos θ is the cosine of the angle θ blw the two vectors.

A.B = Dot product of A and B

||A|| ||B|| - magnitude of vectors A & B.

$$\frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Ex:- Doc 1 : The quick brown fox

Doc 2 : The quick brown dog

vectors :- "The" "quick" "brown" "fox" "dog"

$D_1 = [1, 1, 1, 1, 0]$

$D_2 = [1, 1, 1, 0, 1]$

$A \cdot B = D_1 \cdot D_2 = (1 \cdot 1) + (1 \cdot 1) + (1 \cdot 1) + (1 \cdot 0) + (0 \cdot 1)$

$= 1 + 1 + 1 + 0 + 0 = 3$

$\|D_1\| = \sqrt{1^2 + 1^2 + 1^2 + 1^2 + 0} = \sqrt{4} = 2$

$\|D_2\| = \sqrt{1^2 + 1^2 + 1^2 + 0^2 + 1^2} = \sqrt{4} = 2$

$= \frac{3}{2 \cdot 2} = \frac{3}{4} = 0.75.$

2. Jaccard similarity

It compares the intersection and union of two sets. It is useful when we want to measure the similarity blw 2 documents or a query and document.

Formula :- Jaccard similarity = $\frac{|A \cap B|}{|A \cup B|}$

Ex:- Doc 1: {"data", "mining", "machine", "learning"}

Doc 2: {"data", "science", "machine", "intelligence"}

so • $A \cap B$ = {"data", "machine"} → size = 2

• $A \cup B$ = {"data", mining", machine", "learning", "science", "Intelligence"} = 6

Jaccard = $\frac{2}{3} = 0.33$

3. Dice similarity

Dice coefficient is another measure of set similarity, but gives more weight to the intersection.

Formula = Dice similary = $\frac{2 \times |A \cap B|}{|A| + |B|}$

Ex:- using the same sets

• $|A \cap B| = 2$

• $|A| = 4, |B| = 4$

Dice = $\frac{2 \times 2}{4 + 4} = \frac{4}{8} = 0.5$

Role in IRS

- Helps to rank documents from most to least relevant.
- useful in models like vector space, probabilistic & Neural IR
- Supports fuzzy searching, and partial matching.

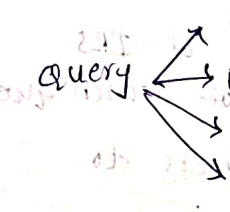
Ranking :-

The ranking is the process of sorting document that match a query in order to display the most relevant results to the user ~~first~~ query.

Ranking approach :-

dataset
↓
Information system
↓
Information Retrival
↓
IR query processing
↓
IR Retrival methods and Ranking process
↓
Evaluation of ranking & retrieval results.

calculating similarity :-

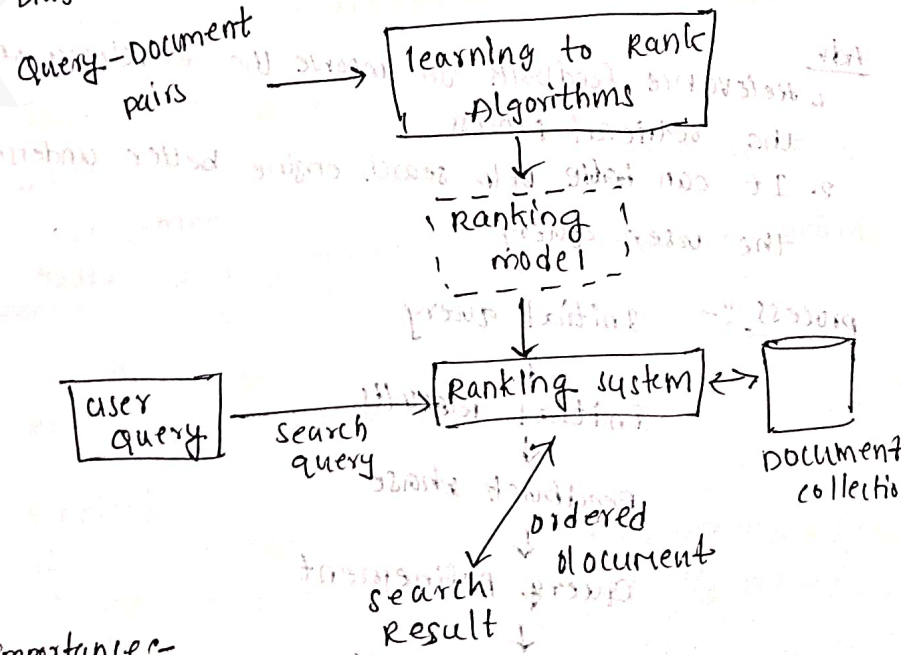


Ranking

D4 — High priority
D2
D1
D3 — low priority

steps :- Query understanding
↓
Document matching
↓
Relevance scoring
↓
Ranking Algo

Diagram :-



Importance :-

- Improve precision and recall
- Enhance user satification
- Helps in relevance-based filtering
- Used in web search engines, digital libraries

Relevance Feedback.

- Relevance feedback is a feature of IRS that improves the effectiveness of search queries.
- It is a mechanism that allow users to provide feedback on the relevance of returned results to their search query
- This feedback is then used to refine and improve furture search results.
- Relevance feedback can be explicit or implicit depending on how the information is collected from the user.

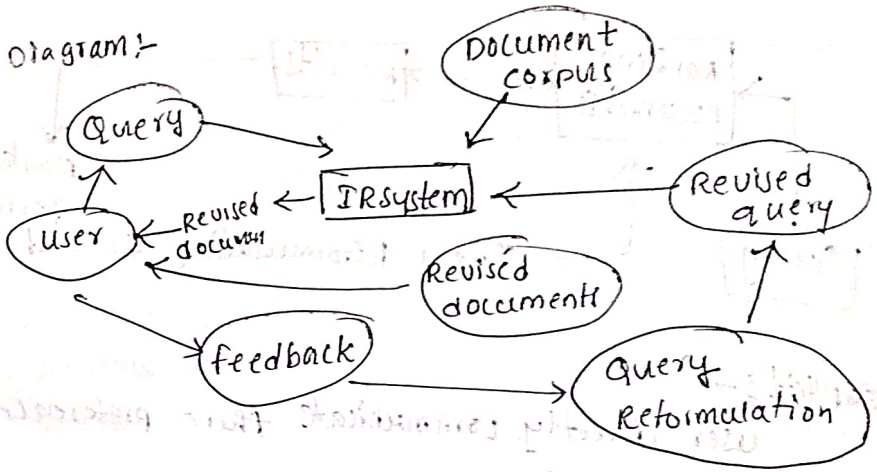
Adv

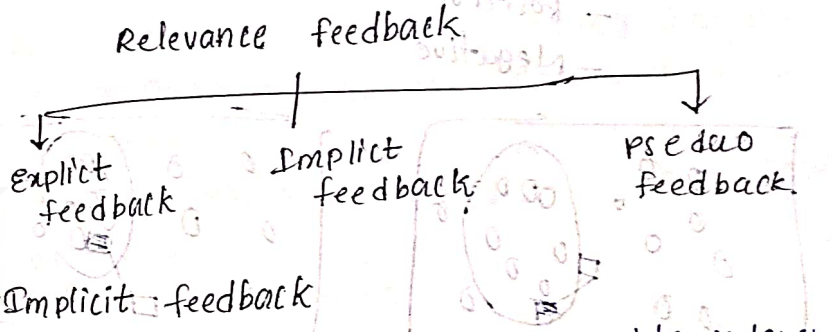
1. Relevance feedback can improve the efficiency of the retrieval process.
2. It can ~~halfe~~ help search engine better understand the user query

process :-

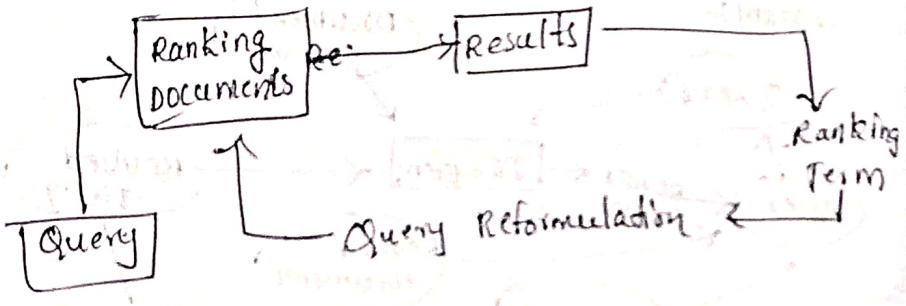
Initial query
↓
Initial Results
↓
Feedback phase
↓
Query Refinement
↓
Revised Result.

Diagram :-

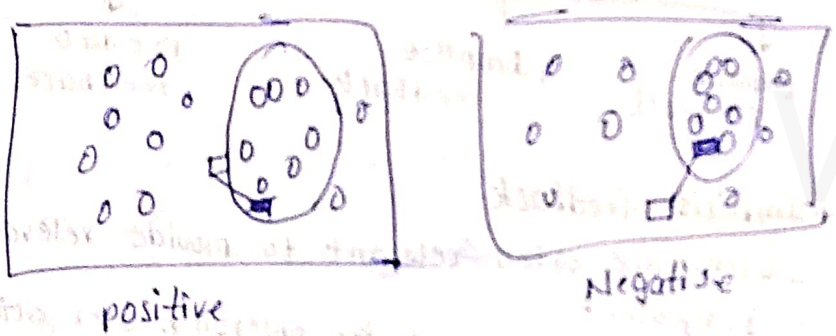




- Implicit feedback
  - Users are often relevant to provide relevance judgement
  - user :- It's gathered by observing user actions and behaviour during their interaction with system.
  - Ex :- Time spending on a document
    No. of click.
- Explict
  It is a method of providing feedback about relevance of search results by directly asking users.
- psendo :-
  - Ran the original query
  - ranking the document
  - Assume top documents psendo relevant
  - construct new query representation
  - Run it, compare rankings.



explicit :-
user directly communicate their preferences
- Positive
- Negative



- The oval represents the items that are retured from the query.
- The solid box is logically where the query initally.
- The hallow box is the query modified by relevance feedback.

Selective Dissemination of Information (SDI)

It is a service that automatically delivers information to users based on their interests and preferences.

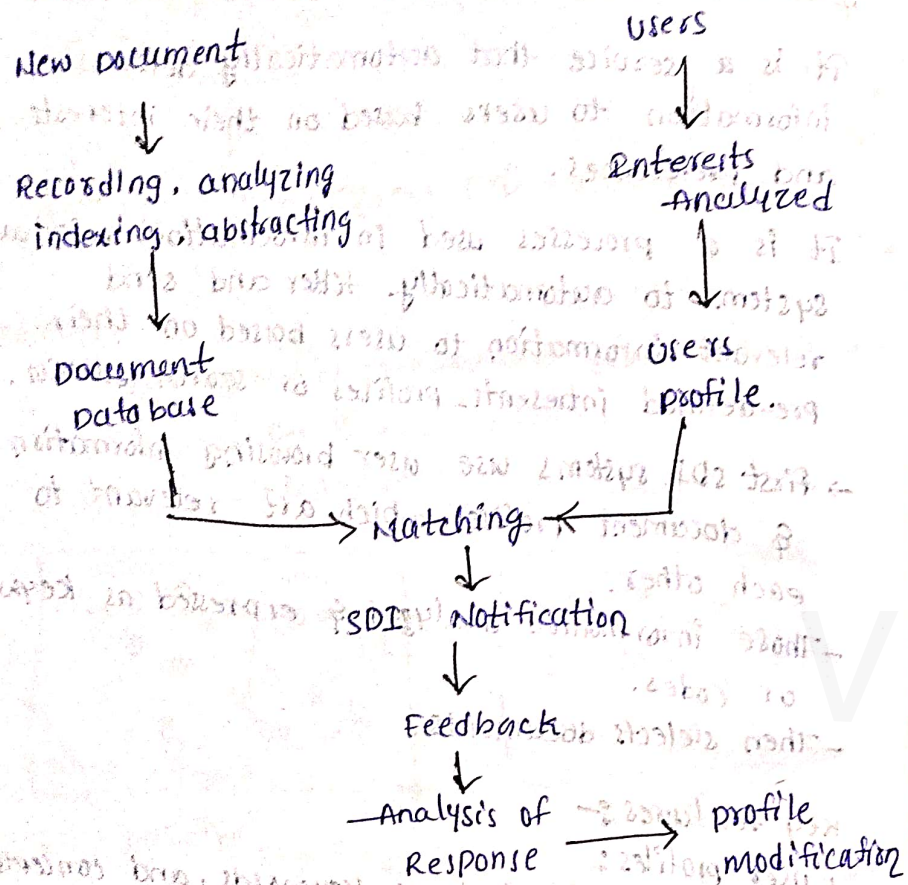
It is a processes used in information retrival systems to automatically filter and send relevant information to users based on their pre-defined intresents, profiles or search criterion.

→ first SDI systems use user browsing information & document profiles which are relevant to each other.
- Those information analyzed & expressed as keywords or codes.
- Then selects documents.

Key features :-

1. User profiles :
   - contains search terms, keywoods, and concepts representing a users info needs.
2. Document Input.
   Newly added documents or data streams.
3. Matching Engine :-
   . compares user profiles with incomming document using similarity measures or Boolean logic
4. Dissemination
   - Relevent documents are delivered to the users inbox, email or dashboard.

SDI Diagram



models :-

1. pull :-
   It is a user-initiated approach, where the user sends a query or request to the system to retrive information.

2. push model:
   It is a system-initiated approach, where the system automatically delivers relevant info. to the user based on their stored interests or profile.

Benfits of SDI

- Increase user satisfication & engagement
- Reduce information overload
- Improve decision-making by providing timely info

Limitations :-

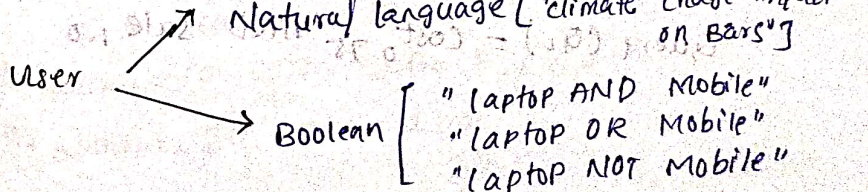
- Needs accurate and updated user profiles
- Risk of missing relevant data if profiles are too narrow
- Complex to implement in multimedia or large scale systems.

Weighted Searches of Boolean Systems

Boolean systems are classical information Retrival (IR) models that use Boolean logic operator like AND, OR, and NOT to retrive relevant documents based on exact term matches.

A weighted Boolean search assigns importance weights to search terms in a Boolean query to influence the ranking of retrieved documents.

The two major approaches to generating quies are Boolean and natural language.



Query :-

LAPTOP AND MOBILE ⟶ IRS ⟶ Results must contain Laptop and Mobile both, cause very few results.

LAPTOP OR MOBLE ⟶ IRS ⟶ Results will contain either Laptop or Moblie, causes too many irrelevant results.

LAPTOP AND MOBILE ⟶ Laptop, AND Mobile,

LAPTOP AND MOBILE ⟶ Laptop, AND Mobile.5

Laptop AND Mobile ⟶ Laptop AND mobile.

Ex:-

	computer	program	cost	sale
D_1	0	4	0	8
D_2	0	2	0	0
D_3	4	0	2	4
D_4	0	6	4	6
D_5	0	4	6	4
D_6	6	0	4	0
D_7	0	0	0	0
D_8	4	2	0	2

step 1- Query

Query (Q_1) = computer$_{1.0}$ OR program$_{0.333}$

Query (Q_2) = cost$_{0.75}$ AND sale$_{1.0}$

Q. step 2 :- strict Interpretation

- Q_1 = computer OR program
- Q_2 = cost AND sale.

Q_1 strict interpretation = $(D_1, D_2, D_3, D_4, D_5, D_6, D_8)$

Q_2 strict interpretation = (D_3, D_4, D_5)

step 3 :- Invariant set (sure documents)

Q_1 invariant = (D_8) Q_2 invariant = (D_3, D_4, D_5)

step 4 :- optional set

- Q_1 optional = $(D_1, D_2, D_3, D_4, D_5, D_6)$
- Q_2 optional = (D_1, D_8)
- Q_1 optional = $(D_1, D_2, D_3, D_4, D_5, D_6) \rightarrow [333 * 6]$ $= [1.99] = 2$ doc
- Q_2 optional = $(D_1, D_8) \rightarrow [1 - .75 * 2] = [0.5] = 1$ doc.

step 5 :- Calculate centroid w.r.t to invariant set.

centroid $(Q_1) = (D_8) = (4, 2, 0, 2)$

centroid $(Q_2) = (D_3, D_4, D_5) =$

$1/3\,(4+0+0,\ 0+6+4,\ 2+4+6,\ 4+6+4)$

$= 1/3\,(4, 10, 12, 14)$

step 6 : Calculate similarity between centroid and doc

Q_1 optional = $(D_1, D_2, D_3, D_4, D_5, D_6) = 2$ DOC

Q_2 optional = $(D_1, D_8) = 1$ DOC

centroid $(Q_1) = (D_6) = (4, 2, 0, 2)$

centroid $(Q_2) = (D_3, D_4, D_5) = 1/3\,(4, 10, 12, 14)$

similarity (centroid$_{Q_1}$, D_1) $= (0+8+0+16) = 24$

similarity (centroid$_{Q_1}$, D_2) $= (0+4+0+0) = 4$

similarity (centroid$_{Q_1}$, D_3) $= (16+0+0+8) = 24$

similarity (centroid$_{Q_1}$, D_4) $= (0+12+0+12) = 24$

similarity (centroid$_{Q_1}$, D_5) $= (0+8+0+8) = 16$

similarity (centroid$_{Q_1}$, D_6) $= (24+0+0+24) = 24$

similarity (centroid$_{Q_2}$, D_1) $= 1/3\,(0+40+0+112) = 1/3\,(152)$

similarity (centroid$_{Q_2}$, D_8) $= 1/3\,(16+20+0+28) = 1/3\,(64)$

Final result :-

- For Q_1, two additional items are added to the invariant set $(D_8) + (D_1, D_3) = (D_1, D_3, D_8)$

- For Q_2, one addition items are added to the invariant set $(D_3, D_4, D_5) + (D_1) =$

$= (D_1, D_3, D_4, D_5)$

- Advantages

* Introduce ranking in boolean
* Helps reflect user priority among terms
* Improves precision by showing most relevant documents first
* More flexible than standard Boolean queries.

Introduction to Information visualization

Information visualization is the practice of representing data and information in graphical or visual format, enabling users to see, understand, and interact with large and complex datasets more effectively.

Benfits :-

- Reduce time to interpret large result sets
- Reveals document - term relationships
- Helps users refine queries.
- Makes information retrieval interactive & engaging
- Improves efficiency.

Cognition and Perception

- Cognition :-

process of thinking, understanding, learning

It refers to the mental process involved in acquiring and understanding knowledge, including attention, memory, problem-solving etc.

In visualization, cognition is concerned with how users:

- Interpret visual structures
- Retain information
- Make decisions based on visual cues.

In the content of Information visualization, cognition and perception play a central role in designing visual interfaces that help users process, understand, and analyze information effectively.

These concepts come from cognitive psychology and human-computer interaction (HCI) and guide how visual representation of data are created in IRS.

Perception:-

How we see

It is ability to see, hear or became aware of something through the senses, especially sight; It affects how user notice, group & prioritize visual information.

perception allows users to:-

- Recognize shapes, colors, patterns
- Detect differences in size, orientation & motion
- Understand groupings in visual data.

~~Aspects of visualization process.~~

Ex:- PARA A → with text

PARA B → text with key underlined

PARA C → the key points coloured.

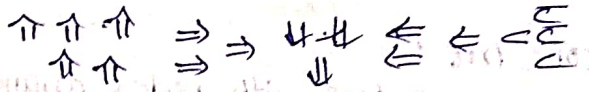
And also by using WIMP

WIMP stands for windows, icons, menus, pointer.

4. Aspects of the visualization process

a. Preattentiv processing

- a low-level, unconscious visual process
- Detects borders, shapes & color changes almost instantly
- useful for grouping in document clustering or highlighting



b. Object Rotation & symmetry

- Rotated objects are harder to recognize
- Vertically aligned items are easier to process

c. color usage

key for classification & emphasis

Attributes: Hue, saturation, Lightness.

Information Visualization Technologies

These are systems and techniques designed to help users understand large volumes of search results by visually representing information.

Main Goals :-

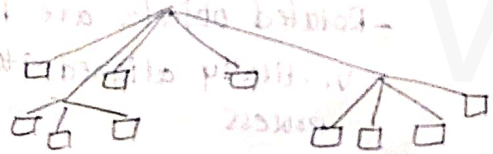
1. Document clustering
   - visually group documents based on content
   - e.g showing documents as clusters in a scatter plot or 3D map.

2. Search statement analysis.
- Help users understand why documents were retrived especially when complex ranking or query expansion is used.
e.g:- Visualizing term contributions using bar charts or heatmaps.

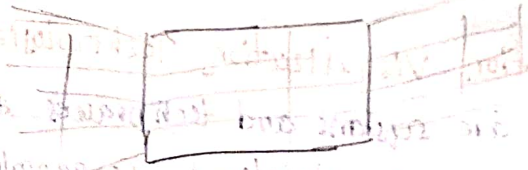
Visualization Techniques & Tools

A. Hierarchical & 3D structures:

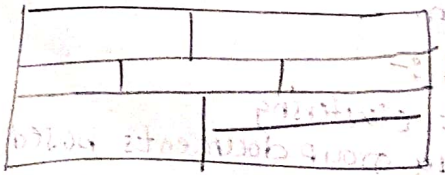
- Cone Tree :-
A 3D tree with nodes arranged in a cone. parent nodes rotate to the front for exploration.



- Perspective wall:-
shows the focus area in the center with side faded into background.



- Tree Maps: Space-filling technique using nested retangles based on data hierarchy



B. Clustering & spatial layouts

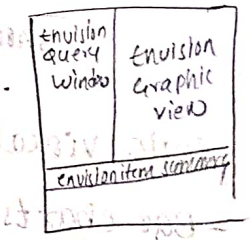
- Scatterplots.
plot documents in 2D or 3D based on context similarity

scatterplots

- Semantic landscapes
Elevation, valleys & hills represent topic density

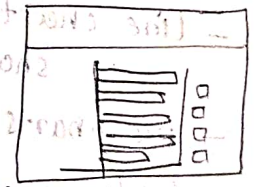
C. Term Relationship & query effect.

- Envision system:
shows query terms, documents and relevance in scatterplots.



Envision

- Veerasamy-Belkin Bars
Bar heights shows term importance per document

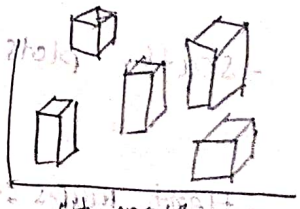


- DCARS:
Histogram of term contribution per item.

DCARS

D. CitySpace & Geometric views.

- cityspace: Uses buildings skyscrapers to show concept importance in 3D.



city space

- WebBook: mimics a virtual library books where documents are browsed like flipping pages.

E. Statistical visualization.

- Courts : plots term frequencies over time.
- Cross field matrix : 2D matrix showing intersections of two fields.
- TileBars :- shows term distribution within a document.

Basic visualization techniques

- Bar charts :- compares quantite across categories.
- Line charts :- show trends over time.
- pie charts :- Represents proportions in a whole
- Histograms :- show frequency distribution of numerical data.
- Scatter plots :- show relationships / correlations btw two variables.
- Heat maps :- show intensity or density over a map.
- Tree maps :- show hierarchical data using nested rectangles.
- Dendrograms - Tree diagrams for showing clustering or taxonomy.

Unit-V

Text Search Algorithms:

- Introduction to Text search Techniques
- Software Text search Algorithms
- Hardware Text search systems.

Multimedia Information Retrival

- Spoken Language Audio Retrieval
- Non-speech Audio Retrieval
- Graph Retrieval
- Imagery Retrieval
- Video Retrieval.

Introduction to Text search Techniques

They are used in Information Retrival systems to efficiently locate relevant documents based on a user's query. These techniques aim to match patterns (query terms) within a large collection of text and retrieve items and that satisfy the users search conditions.

They help:
- Quickly identify relevant items
- Eliminate irrelevant ones
- Improve search performance and user satisfication.

In a textual database three classical text retrival techniques have been defined.

Those techniques are.

1. Streaming (full text scanning)
   - scans the entire database to find matches
   - Useful for exact matches, stop words, or highlighting terms in results.

2. Word Inversion (Inverted Index)
   - Builds a word-to-document index
   - Enables fast lookup by skipping non-matching documents.

3. Multi Attribute Retrival
   - Searches across multiple fields like title, author and body.
   - Provides flexible search capabilites.

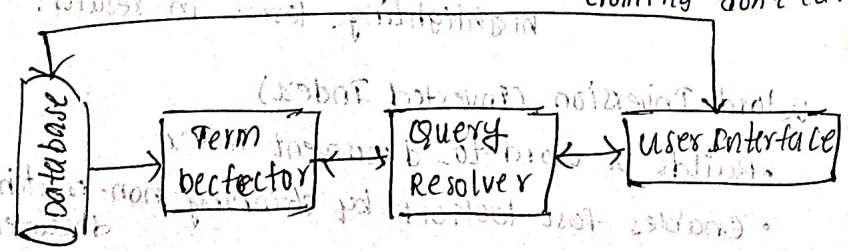
- First we use index mechanism for searching text. Now streaming of text was frequently found in the system as search technique.
- Generally query entered by one user. In text scanning system one or more users enter queries.
- Earlier systems used hardware-based text streamers due to cpu limitations. But modern systems mostly rely on software-based search due to increased computing power.

Drawbacks of early techniques:-

- manual - slow, error-prone
- No partial matching or ranking
- Hardware - costly. Outdated.

Example of limits of index searches are:-

- Search for stop words
- Search for exact matche when stemming is performed
- Search for terms that contain both leading and trailing "don't cares".



Text streaming Architecture.

- Term detector is a specially contains all of the terms being searched & also contains logic btw the items.
  It will give output to the query resolver It process finally.
- Query resolver accept search statements from the users & extract the logic & search terms & pass the search terms to the detector.
- It accepts results from the detector & determines which queries are satisfied by the item & possibility.
  Finally it pass the info to the user interface that will continually updating search status.

There are two approaches to the data stream.

1. Complete database is being sent to the detector.
2. Random retrieved items are being passed to the detectors.

Many of thlw & slw text searches use finite automata as a basis for their algorithms

I - set of I/p symbols from alphabet
S - set of states
P - set of productions
S_0 - Initial state
S_f = One or more final state.

Software Text Search Algorithms

Software Text search Algorithms are software based techniques used to locate specific patterns (search terms) within a body of text. These are typically used when full text is available in memory and no hardware support is required.
There are four major techniques associated with software text search.

Slw streaming techniques. Items read into memory & then the algo is applied.

1. Knuth-morries-pratt (KMP)

KMP is a powerful pattern matching algorithm that reduces time complexity from O(n x m) to O(n+m) using the LPS table.
It is efficient for searching long patterns in large text.

Avoids rechecking matched characters using a partial match table (prefix function)

- Preprocessing takes O(m), matching takes O(n)
- Time complexity : O(n+m)
- Good for predictable and long patterns searches.

ex:- a b c d a bc

prefix:- a, ab, abc, abcd
suffix:- c, bc, abc, dabc } abc

pattern
a b a b d

string:- a b a b c a b c a b a b d
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

pattern a b a b d
0 0 1 2 0

π or lps

P1 : a b c d a b e a b f
0 0 0 0 1 2 0 1 2 0

P2 : a b c d e a b f a b c
0 0 0 0 0 1 2 0 1 2 3

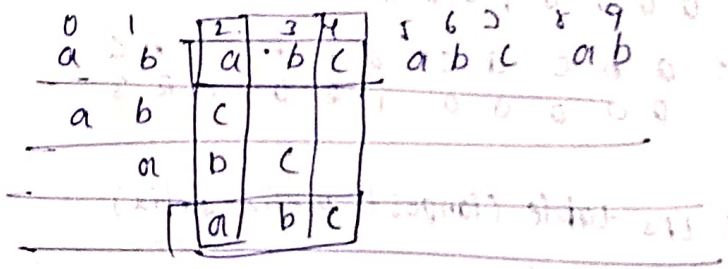
Steps:-
1. Build LPS table (longest prefix sufix)
2. perform search
   i. start with two pointers
      - i for text
      - j for pattern
   ii. Compare text[i] with pattern[j]:
      - If they match move both i++ & j++
   iii. if j reaches end of pattern
      - pattern found at position i-j
      - Reset j = lps[j-1] to look for more matches
   iv. If mismatch
      - If j≠0 backtrack using LPS → j = lps[j-1]
      - If j==0 move i++ only (start next search)

2. Brute force Algorithm :-

- compares pattern with the text character by character.
- If mismatch occurs shift by one and repeat
- Time complexity : worst case → O(n x m)
- Simple but inefficient.
- It is a simplest way to search for patter in text

ex :- Text = a b a b c a b c a b

Pattern = a b c



3. Boyer-moore Algorithm

It is an efficient pattern matching algorithm that searches from right to left and uses smart skipping to avoid unnessary comparisions.

steps :-

1. First construct "Bad Match table"
2. Compare right most character of pattern with given string based on value of bad match table.
3. If mismatch then shift the pattern to the right position corresponding to the value.

- Extremely efficient in practice
- Best case : sub linear time ;
  worst case O(n+m)
- most effective for long patterns and large texts.
- Faster than KMP

Ex :- Text :- WELCOME TO LAB MUG

Bad match table

LA	B	M	U	G	*
5	4	3	2	1	6

0 1 2 3 4 5 6

length - index - 1

L = 6 - 0 - 1 = 5

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

→ WELCOME TO LAB MUG

LAB MUG

* not matched value of m = 2 - shift 2 steps right

→ WELCOME TO LAB MUG

LAB MUG

* again shift 2 steps right

LAB MUG

- Then it is matched.

4. Rabin-Karp Algorithm

- uses hashing for pattern and substring comparision.
- Good for multiple pattern search.
- expected Time : O(n), but worst case can degrade to O(n x m) (due to hash collisions)
- often used in plagiarism detection or searching multiple patterns.

Hardware Text search systems :-

While software based text search algorithms are widely used, they face performance limitations when:

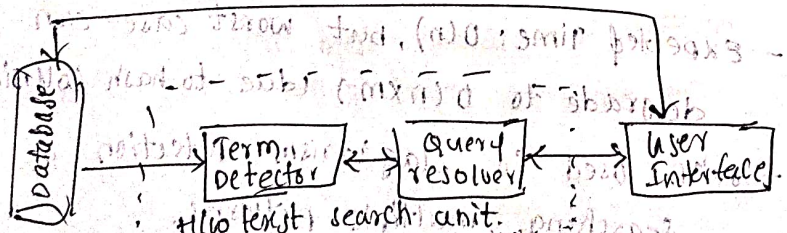
- Preprocessing many search terms simultaneously
- Dealing with large-scale text databases
- Or facing I/O speed bottlenecks.

To over come these, Hardware Text search systems were developed to

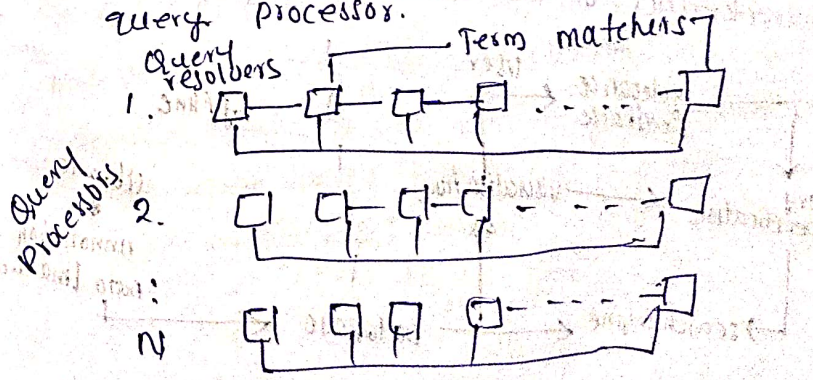
- Offload search processing from the CPU.
- Improve speed and scalability.
- Eliminate the need for index structures.

→ It is a specialized machine that performs searches on text data.

- First machine to perform the searches & pass the results to the main computer which supported the user interface & retrived of hits.



- One of the earliest Hw text searching search unit was the Rapid search. It developed by General Electric.
- The machine consisted of a special purpose search unit where a single query was passed against a magnetic tape containing the documents.
- The next developed Associative file processor (AFP)
  - OSI following a different approach developed by High speed text search (HSTS)
  - GE redesigned their Rapid search machine into the GESCAN unit
  - Here uses Text Array processor (TAP) an array of four to 128 query processors
- The text is loaded into the cache & searched by the query processors.
- Each query processor is independent and can be loaded at any time.
- A complete query is handled by each query processor.



A query processor work two operations in parallel

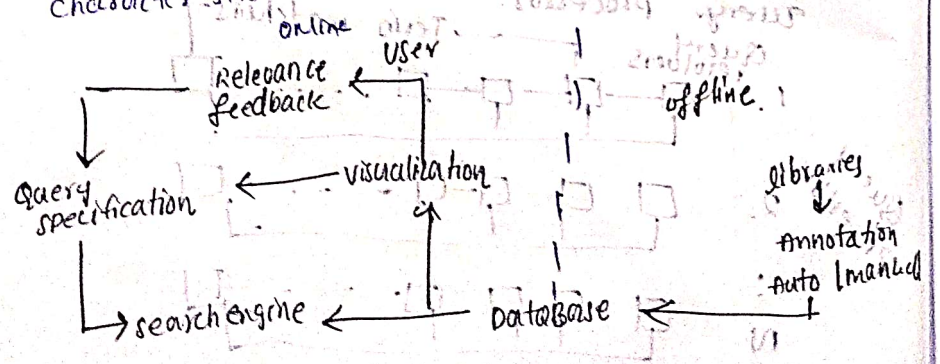
→ Matching query terms to input text & Boolean logic resolution.

Term matching is performed by a series of characters cells each containing one character.

Multimedia Information Retrieval (MMIR)

multimedia Information Retrival refers to searching and retrieving relevant multimedia content like Audio, image, Graphs, videos, Animations from large databases, using content-based or metadata-based technique.

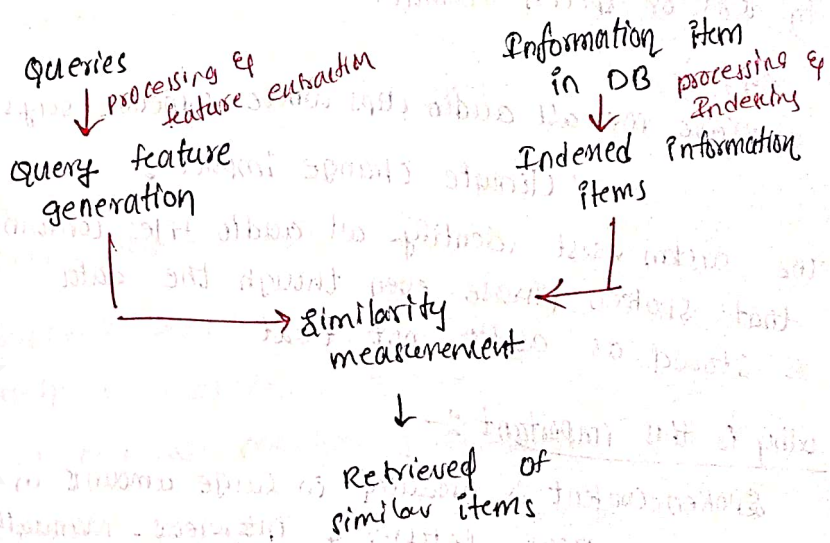
- It aims at extracting semantic information from multimedia data sources.
- Data sources like audio, video, image
- The architecture of multimedia IRS depends on the characteristics of the multimedia data.



In multimedia Information Retrieval

- extracts semantic
- stores, indexes, searches & data.
- → analyzes & searches digital documents.

Overview of MIRS



Components of MMIR system

1. Data collection
2. Feature extraction
3. Indexing
4. Query Interface
5. Matching & Ranking
6. Retrieval output.

MMIR = multimedia + Relevance + search

- Involves analyzing and retrieving non-textual content
- Uses content-based feauters & metadata
- supports muti-modal queries

Spoken language Audio Retrieval

Spoken language audio Retrieval (SLAR) is the process of retrieving spoken-word audio recordings based on a users search query - which could be in text or speech formal.

eg:-
"Give me all audio clips where someone says 'climate change impact'."

The system must identify all audio files containing that spoken phrase, even though the data is stored as audio - not text.

Why is this important:-

Spoken content is produced in large amounts in areas like news, lectures & interviews. Manually reviewing it takes too much time, so smart systems are needed to quickly find important parts.

How it works:-

step 1:- Speech Recognition (ASR)
Automatic speech recognization (ASR) is used to convert speech into text (called transcription)
This acts as a bridge between the spoken data and traditional text retrieval

Audio → [ASR] → Text → [Indexing & Retrival]

step 2:- Indexing the text.
The text is organized using eco-friendly methods

step 3:- Search:
A user types or says a query, which is matched against the indexed text.

step 4: Get Audio:-
The system finds and returns audio clips and timestamps where the query is spoken.

Advantages:-
- Access spoken content without listening to hours of data.
- Supports search by speech and search-by text
- Helpful in archieving, journalism, legal, education
- Enables fast retrival of relevant spoken conent.

Challenges:- ASR errors, semantic Gap, multiple speaker Nosiy audio, Language & Dialects.

* Spoken Audio → ASR → Text → search → Retrieve audio.

Non-Speech Audio Retrival

Unlike spoken language retrieval (which relies on converting speech to text), non-speech audio retrival is about searching sounds that are not speech-like music, alarms, animal calls, or environmental sounds (rain, wind, traffic etc.

How it work:

1. first you extract audio features from the sound
   -pitch, Rhythm, Timbre, energy, spetical features.
2. These features are then stored in an index.
3. when the user submits a query - for example by giving an example sound or describing it - the system tries to match those features to sounds in its database.
4. It then returns relevant results:
   e.g:- "find me all files with sirens".

Techniques :- -query by humming / singing
-Audio fingerprinting

Applications
- music retrival, Audio surveillence, wildlife monitoring

challenges:-
- Audio signals are more variable and harder to classify than text
- Background noise can interfere
- lack of standard "words" to represent many sounds.

Graph Retrieval:-
It is about searching and retrieving info also from data that is organized as graphs - that is collections of nodes (vertices) and edges (links) between them.

How does graph retrieval work:-
1. First a graph database or index is built that stores nodes, edges and their attributes.
2. when a user gives a query (often a subgraph pattern) the system:
   -matches the pattern -ranks results
   -retrieves graphs (or subgraphs) that match

for e.g:-
- "find all subgraphs similar to the triangle shape".

common Techniques :-
Subgraph isomorphism, Graph similarity matching, Graph embeddings, Graph indexing

Applications:-
-Drug discovery, social network analysis
-fraud detection -web structure mining

challenges:-
-Graphs are computationally hard to search.
-Query performance on very large graphs.

Image Retrieval

Imagery retrieval is about retrieving relevant images from a large collection, using either text label (TBIR) or visual features (CBIR), with growing interest in combining the two or for better results.

Types of Image Retrieval

There are 2 major approaches:

1. Text-Based Image retrival (TBIR)
   - Images are described by manually assigned keywords, captions, tags
   - search uses these textual labels
   - limitation: subjective, inconsistent, labor intensive.

2. Content-Based Image retrival (CBIR)
   - Images ~~are~~ are indexed using their actual visual content.
     (e.g. color, texture, shapes, spatial layout)
   - No manual tagging needed.
   - example: query by example - you give an example image, and system retrieves visually similar images.

How does CBIR work:-

1. Feature extraction
   The system analyzes images to extract low-level features like
   - color histograms
   - texture patterns, shapes, edges.

2. Indexing
   - features are organized into an efficient data structure (like Inverted file structure) for fast searching.

3. Query processing
   - A user provides a query (keywords, or even an example image)

4. Similarity matching
   - The system compares the query features with stored image features using distance/similarity measures.

5. Ranking & Retrieval
   - Images are ranked by similarity to the query and presented to the user.

challenges:-
- Semantic gap
- variation — lighting, viewpoint, quality
- subjectivity — "similar" is subjective.

Applications:-
- Digital photo achievers
- medical image retrival
- mesum and heritage
- E-commerce
- surveillance.

Video Retrival

Video retrival is the process of searching for and retrieving relevant video segments from a large collection based on a users query.

unlike text or image, video is more complex because it has:-
- visual (frames, scences)
- audio (speech, music, sound effects)
- temporal information (sequence motion)

How it works :-

1. Video segmentation
   - Divides the video into meaningful shots or scenes
     - A shot is an unbroken sequence of frames ~~from~~ from one camera.
     - A scence is group of related shots.

2. Feature extraction :-
   - Extracts visual features (color, shape, motion patterns)
   - Extracts audio features (speech recognition, music dection)
   - Sometimes includes text (closed captions, superimposed text)

3. Indexing
   - Stores features, metadata, and scence boundaries is an efficient data structure to allow fast searching.

4. Query formulation :
   - Users can search by
     - Keywords (e.g. "football goals")
     - Example video (query-by-example)
     - sketch (sketch based retrieval)
     - Audio clues (humming a song)

5. Similarity matching
   - Compares features of the query to features in the stored videos
   - Ranks videos by how well they match.

6. Retrieval and display.
   - Retrieves ranked results.
   - Sometimes displays key frames so they user can quickly browse them

Challenges :-
- large data volumes
- semantic gap
- Temporal structure
- subjectivity

Applications
- News archives, sports highlights, video-on-demand
- movie sence searches
- Eduction / lecture retrieval systems.

Video retrieval is about intelligently searching large video collections using visual, audio and sometimes textual clues to help users find the right content quickly.