Lalitha Priya Bijja
101168225

CSC 785 - INFORMATION STORAGE AND RETRIEVAL

1. Information Retrieval Vs Database Management System

Information Retrieval	Database Management System
IR is about finding documents of an unstructured data (usually text) from a corpus, that satisfies the need.	DBMS, on the other hand, is designed for the structured storage, management, retrieval, and manipulation of data. It is used to maintain and organize structured data in a way that allows for efficient querying and transaction management.
IR deals with unstructured or semi-structured data where the relationships between data elements are not predefined. Egs: Documents or web pages are data sources in IR.	DBMS deals with structured data organized in tables, where data elements have predefined relationships. Eg: Relational databases (data is stored in rows and columns).
IR systems typically use keyword-based queries to search for documents or information. Queries can be more flexible and natural language-oriented.	DBMS uses structured query languages (e.g., SQL) to retrieve data based on specific criteria. Queries are more rigid and focused on structured data attributes.
IR systems are more flexible and can handle a wide range of data types, including unstructured text, images, audio, and video.	DBMS is less flexible in terms of data types and is better suited for structured data. It may not handle unstructured data as effectively.
IR is commonly used in web search engines, content recommendation systems, plagiarism detection, and document retrieval.	DBMS is used in various applications such as e-commerce websites, inventory management, customer relationship management (CRM), and financial systems.

Example:

Scenario: A scientist wants to find research papers on a specific topic, such as "natural language processing in the healthcare industry."

Information Retrieval: The scientist could use an academic search engine like Google Scholar or PubMed to find relevant papers. These search engines index and rank research papers based on their titles, abstracts, keywords, and citations. The scientist could also use natural language queries to refine their search, such as "NLP healthcare review papers published in the last 5 years."

Database Management System: The scientist could also use a database to find relevant research papers. For example, Scopus is a database that indexes millions of scholarly articles, books, and conference proceedings. The scientist could use SQL queries to search the Scopus database for papers that meet their criteria.

2.

Boolean Retrieval model

- The Boolean retrieval model is a simple but powerful model for information retrieval.
- It allows users to construct Boolean queries to retrieve documents that satisfy specific criteria.
- Boolean expressions that combine terms with the logical operators AND, OR, and NOT are known as Boolean inquiries.
- For example, the Boolean query "cat AND dog" would retrieve all documents that contain both the terms "cat" and "dog." The query "cat OR dog" would retrieve all documents that contain either the term "cat" or the term "dog." The query "cat NOT dog" would retrieve all documents that contain the term "cat" but do not contain the term "dog."

The Boolean retrieval model is easy to understand and implement. It is also efficient for processing simple queries. However, there are a few disadvantages:

a) It is difficult to construct complex Boolean queries that accurately capture the user's information need.
b) The results are not ranked according to how relevant they are to the query.
c) The Boolean retrieval model requires exact matches between query terms and document terms, making it inflexible for handling natural language variations, such as synonyms, stemming, or spelling mistakes.
d) It is not suitable for retrieving documents from large collections of unstructured data.

Term-Incidence Document Matrix

- The term-incidence document matrix is a data structure that represents the relationship between terms and documents in a collection.
- It is a binary matrix, where each row represents a term and each column represents a document.
- A cell in the matrix is 1 if the corresponding term occurs in the corresponding document, and 0 otherwise.
- **For example**, the following term-incidence document matrix represents a collection of three documents:

Doc 1 : cat eats mouse
Doc 2 : dog eats cat
Doc 3 : mouse eats cat and dog

Term	**Doc 1**	**Doc 2**	**Doc 3**
Cat	1	1	1
Dog	0	1	1

Disadvantages

a) It can be very large for large collections of documents.
b) It does not support ranked retrieval.
c) It does not support wildcards or other advanced query features.

Inverted Indexing

- Inverted indexing is a data structure that is commonly used to implement Boolean retrieval systems. It is a more efficient data structure than the term-incidence document matrix, especially for large collections of documents.
- A hash table that links phrases to documents is called an inverted index.Each entry in the hash table contains a list of all the documents that contain the corresponding term. The list can also include additional information, such as the number of times the term occurs in each document.
- **For example**, the following inverted index represents the same collection of documents as the term-incidence document matrix above:

Doc 1 : cat eats mouse
Doc 2 : dog eats cat
Doc 3 : mouse eats cat and dog

term	**Documents**
Cat	[1, 2, 3]
dog	[2, 3]

Inverted indexes can be used to efficiently process Boolean queries using a technique called set intersection. To answer the query "cat AND dog," we can simply take the intersection of the lists of documents for the terms "cat" and "dog." This will return a list of all the documents that contain both terms.

Why Inverted Indexing is Chosen

There are various reasons why inverted indexing is preferred over the term-incidence document matrix.

a) **Storage efficiency:** The term-incidence document matrix can be very large for large collections of documents. In contrast, the inverted index is much more compact, as it only stores the documents that contain each term. This can save a significant amount of storage space.
b) **Query processing:** The term-incidence document matrix is not well-suited for processing complex Boolean queries. In contrast, the inverted index is much more efficient for processing Boolean queries, as it allows for efficient set intersection and other operations.
c) **Ranked retrieval:** The term-incidence document matrix does not support ranked retrieval. In contrast, the inverted index can be used to implement ranked retrieval algorithms, such as TF-IDF (term frequency–inverse document frequency).

- Doc 1: new home sales top forecasts
- Doc 2: home sales rise in July
- Doc 3: increase in home sales in July
- Doc 4: July new home sales rise

Term Document Matrix

Terms	Doc 1	Doc 2	Doc 3	Doc 4
new	1	0	0	1
home	1	1	1	1
sales	1	1	1	1
top	1	0	0	0
forecasts	1	0	0	0
rise	0	1	1	1
in	0	1	1	0
July	0	1	1	1
increase	0	0	1	0

1 = presence of a term in document
0 = indicates absence

Inverted Index

Terms	Docs
new	Doc 1, Doc 4
home	Doc 1, Doc 2, Doc 3, Doc 4
Sales	Doc 1, Doc 2, Doc 3, Doc 4
Top	Doc 1
Forecasts	Doc 1
Rise	Doc 2, Doc 3, Doc 4
In	Doc 2, Doc 3
July	Doc 2, Doc 3, Doc 4
increase	Doc 3

3.

a) home AND July – Doc 2, Doc 3, doc 4

b) top OR sales – Doc1, doc2, doc 3, doc 4

4.

O (N+M)
where,
n &m are the size of posting lists

Combining complexities of both intersections
O(N,M) + O(A,B)