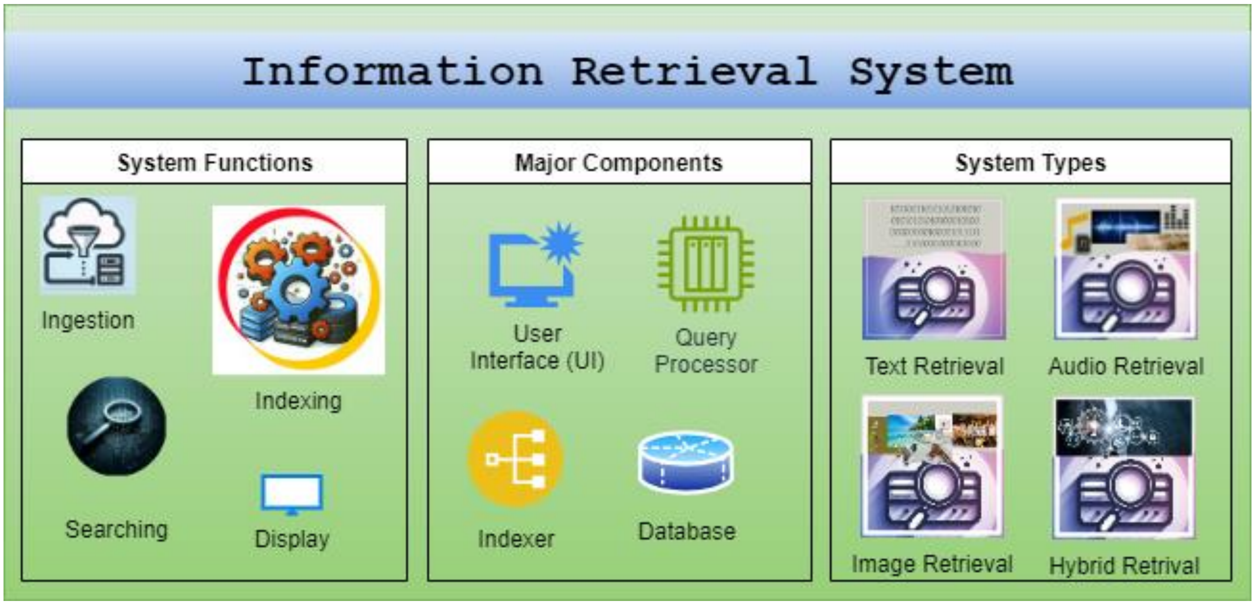
Assignment 01: Information Web Retrieval

Name: KRY Senghort

Group: I5-AMS-B

Exercise 1

1.) **Consider the definition of Information Retrieval (IR) as stated by Salton (1968): "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."**



Answer: To further clarify the definition of Information Retrieval (IR) by Salton (1968): Information Retrieval (IR) is a multidisciplinary field that focuses on how information can be effectively managed and accessed. Let's elaborate on the components mentioned in the definition:

- Structure: This refers to how information is formatted and arranged. Information can be structured (like databases or spreadsheets) or unstructured (like text documents, images, or audio files). Structuring data appropriately is essential for efficient retrieval. For example, in search engines, web pages are structured through metadata, tags, and hyperlinks to make them easier to search.
- Analysis: Analyzing information involves interpreting and understanding its content. In IR, this often means using algorithms and techniques like keyword extraction, natural language processing (NLP), sentiment analysis, and machine learning to discern relevant insights from the data.

Analysis helps refine searches and enhances the accuracy of the retrieval process by recognizing patterns, relationships, or context in the information.

- Organization: This aspect deals with arranging information in a manner that facilitates easy access. For instance, libraries use cataloging systems to organize books by subject, author, and publication date. In digital systems, information is often indexed so that it can be efficiently retrieved based on queries. An example of organization in IR systems is the indexing used by search engines to quickly locate relevant pages for a given search query.
- Storage: Storage refers to how and where information is kept, whether in physical locations like libraries or in digital databases, cloud storage, or other data repositories. In IR, storage systems need to balance capacity with the speed of retrieval. Effective storage strategies ensure that information is preserved in an accessible and efficient way, particularly when handling large volumes of data.
- Searching: Searching is the process by which users seek specific information. In IR, this involves developing search algorithms and interfaces that allow users to enter queries and retrieve the most relevant results. Search engines, for example, rank results based on relevancy, using methods like term frequency, inverted indexing, and ranking models.
- Retrieval: Retrieval refers to the actual process of getting the desired information. Effective IR systems not only find the information but also rank it according to relevance to the user's needs. Retrieval mechanisms include full-text search, Boolean search, vector space models, and probabilistic retrieval models. The goal of retrieval is to minimize irrelevant results and maximize the usefulness of what is retrieved.

In summary, Information Retrieval (IR) encompasses the entire process of managing and accessing information, from how it's structured and stored, to the algorithms and techniques used to search and retrieve it. This field plays a critical role in modern digital environments, enabling the development of search engines, recommendation systems, digital libraries, and other tools that help users find the information they need.

2.) **Identify and explain the key components of this definition in the context of a web search.**

The definition of Information Retrieval (IR), as stated by Salton (1968), highlights several key components that are essential for understanding how IR functions in the context of a web search. Here’s a breakdown of these components and how they apply to web search systems:

- Structure

  - Definition: Structuring involves organizing raw web data into a format that makes it easy to process and search.
  - In the context of a web search: Web pages are structured using HTML, which defines the content and layout of a page. Search engines utilize this structure to identify the main elements of a webpage, such as headings, paragraphs, metadata (titles, descriptions, keywords), and links. These elements are critical in helping search engines understand the content and context of the page for better search results.
- Analysis
  - Definition: Analysis refers to processing and interpreting the information to extract relevant meaning.
  - In the context of a web search: Web search engines analyze both user queries and the content of web pages. This involves several techniques such as natural language processing (NLP) to understand query intent, keyword extraction, and semantic analysis to interpret the relevance of web content. Search engines like Google use sophisticated algorithms to match the meaning of user queries with the content of indexed web pages.
- Organization
  - Definition: Organization refers to categorizing and arranging information in a way that makes it easier to retrieve.
  - In the context of a web search: Search engines use indexing to organize web content. Indexing involves creating a data structure that maps keywords and other relevant content from web pages to the URLs where they appear. This enables efficient searching and ensures that relevant information can be quickly accessed when a user performs a search query.
- Storage
  - Definition: Storage refers to preserving data in a system so it can be retrieved later.
  - In the context of a web search: Web search engines store vast amounts of web data in massive databases, often referred to as search engine indexes. These databases store snapshots of web pages (crawled by bots or spiders) along with relevant metadata. The data must be stored in a way that allows for quick retrieval when a query is made.
- Searching
  - Definition: Searching involves querying a system to locate relevant information.
  - In the context of a web search: The search process begins when a user enters a search query in the search engine. The search engine then scans its index to find pages that match the query, ranks these pages based on relevance, and presents them to the user. The

search algorithm uses various signals like keyword matching, page quality, and user behavior (e.g., click-through rates) to determine which pages should rank higher in the search results.

- Retrieval
    - Definition: Retrieval involves delivering the most relevant information in response to a query.
    - In the context of a web search: Once the search engine has identified relevant web pages, it retrieves and ranks them based on their relevance to the user query. The results are presented in an ordered list (typically a search engine results page, or SERP), with the most relevant or authoritative pages at the top. Search engines use relevance-ranking algorithms that weigh factors like content quality, keyword relevance, user engagement, and the credibility of the website to retrieve and display results.

In summary of Key Components in Web Search:

- Structure: Web pages structured in HTML help search engines interpret content.
- Analysis: Search engines analyze both user queries and web content using algorithms.
- Organization: Web content is indexed for efficient retrieval.
- Storage: Crawled web pages are stored in databases (indexes).
- Searching: Users query search engines to find relevant web content.
- Retrieval: The most relevant pages are retrieved and ranked to meet user intent.

Each of these components plays a crucial role in making web search effective, allowing users to find the information they need from vast amounts of data on the internet.

Exercise 2

1. **Write a Python function that simulates a simple information retrieval system. The function should take a list of documents (strings) and a search query (string) as inputs and return a list of documents that contain the search query.**

```python
def simple_information_retrieval(documents, query):
    """
    Simulates a simple information retrieval system.

    Parameters:
    documents (list of str): List of documents (strings).
    query (str): The search query (string).

    Returns:
    list of str: List of documents that contain the search query.
    """

    # Convert the query to lowercase to make the search case-insensitive
    query = query.lower()

    # List to store the documents that contain the search query
    matching_documents = []

    # Iterate through each document in the list
    for doc in documents:
        # Check if the query is present in the document (case-insensitive)
        if query in doc.lower():
            matching_documents.append(doc)

    return matching_documents


# Example usage:
documents = [
    "Information retrieval is a key field in computer science.",
    "This document discusses machine learning and its applications.",
    "Data retrieval and storage are crucial for modern information systems.",
    "Information retrieval systems help users find relevant documents."
]

query = "information retrieval"

result = simple_information_retrieval(documents, query)

# Display the result
print("Documents containing the query:")
for doc in result:
    print(f"- {doc}")
```

Output:

```
Documents containing the query:
- Information retrieval is a key field in computer science.
- Information retrieval systems help users find relevant documents.
```

2.) Use the in operator to check for the presence of the query in each document. Make sure the search is case-insensitive.

Document List:

documents = [
"A swift auburn fox leaps over a sleepy canine.",
"Data search heavily relies on information retrieval.",
"Text comprehension is enhanced by natural language processing.",
"The sleepy dog relaxes in the warm sun"]

```
def simple_information_retrieval(documents, query):
    """
    Simulates a simple information retrieval system using the `in` operator.
    Parameters:
    documents (list of str): List of documents (strings).
    query (str): The search query (string).

    Returns:
    list of str: List of documents that contain the search query.
    """

    # Convert the query to lowercase to make the search case-insensitive
    query = query.lower()

    # List to store the documents that contain the search query
    matching_documents = []

    # Iterate through each document in the list
    for doc in documents:
        # Check if the query is present in the document (case-insensitive)
        if query in doc.lower():
            matching_documents.append(doc)

    return matching_documents


# Document list
documents = [
    "A swift auburn fox leaps over a sleepy canine.",
    "Data search heavily relies on information retrieval.",
    "Text comprehension is enhanced by natural language processing.",
    "The sleepy dog relaxes in the warm sun."
]
# Search query
query = "sleepy"

# Perform the search
```

```
result = simple_information_retrieval(documents, query)


# Display the result
print("Documents containing the query:")
for doc in result:
    print(f"- {doc}")
```

Output:

```
Documents containing the query:
- A swift auburn fox leaps over a sleepy canine.
- The sleepy dog relaxes in the warm sun.
```