LP-III (ISR LAB ORAL QUESTION)

https://chatgpt.com/share/67169950-d310-8011-9844-7705a7929917

ASSIGNMENT-1

1. Difference Between Data Retrieval and Information Retrieval

Aspect	Data Retrieval	Information Retrieval
Objective	Retrieves exact data stored in databases	Retrieves relevant information from unstructured sources
Input	Structured queries (SQL, NoSQL)	Keywords, natural language queries
Data Type	Structured (tables, records)	Unstructured (text, multimedia)
Result	Exact match of records	Relevant documents or content
Application	Databases	Search engines, knowledge management systems

---

2. What is a Conflation Algorithm?

A **conflation algorithm** is used to reduce multiple forms of a word to a single root or base form, primarily for text processing and indexing. It aims to improve search efficiency by treating variations of a word as one.

Steps:

1. **Word Segmentation**: Break down text into individual words.
2. **Suffix Removal**: Remove common word endings (e.g., -ing, -ed).
3. **Prefix Removal**: Remove common word prefixes (e.g., un-, dis-).
4. **Stemming**: Reduce the word to its root form.
5. **Normalization**: Apply rules for standardization (lowercasing, singularizing).

---

3. What is Luhn's Idea?

Luhn's Idea focuses on **automatic text summarization** using statistical methods. It identifies the most important sentences by analyzing word frequency and positioning.

Sections:

- **Word Frequency Analysis**: Identifies frequently occurring words in a document.
- **Threshold Filtering**: Sets frequency limits to determine which words are considered important.
- **Cluster Formation**: Groups significant words that appear close to each other in sentences.
- **Summary Extraction**: Extracts sentences containing key clusters to form a summary.

---

4. What are Stopwords?

Stopwords are commonly used words (e.g., "the", "is", "and") that are filtered out during text processing because they carry little to no significance in search queries or indexing.

---

5. What is a Document Representative?

A **document representative** is a compact representation of the document's content, often in the form of keywords, key phrases, or vectors, used to summarize and retrieve relevant documents effectively.

---

6. Explain Indexing, Exhaustivity, and Specificity

- **Indexing**: The process of creating a searchable index from documents to facilitate efficient retrieval.
- **Exhaustivity**: Refers to the extent to which all topics or keywords of a document are indexed.

  Exhaustivity=Number of indexed termsTotal relevant terms in document\text{Exhaustivity} = \frac{\text{Number of indexed terms}}{\text{Total relevant terms in document}}Exhaustivity=Total relevant terms in documentNumber of indexed terms

  Example: A document has 10 relevant topics, but only 7 are indexed, so Exhaustivity = 7/10 = 0.7

- **Specificity**: Measures the degree to which the indexed terms reflect the specific content of the document.

  Specificity=Number of unique, detailed termsTotal indexed terms\text{Specificity} = \frac{\text{Number of unique, detailed terms}}{\text{Total indexed terms}}Specificity=Total indexed termsNumber of unique, detailed terms

  Example: If out of 7 indexed terms, 5 are highly specific, then Specificity = 5/7 = 0.71

---

7. Five Commonly Used Measures of Association in Information Retrieval

1. **Jaccard Coefficient**: Measures similarity between two sets.

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

2. **Cosine Similarity**: Measures the cosine of the angle between two vectors.

$$\text{Cosine}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

3. **Dice Coefficient**: Measures similarity based on shared elements between sets.

$$\text{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

4. **Pointwise Mutual Information (PMI)**: Measures how much information two variables share.

$$\text{PMI}(x, y) = \log \frac{P(x, y)}{P(x)P(y)}$$

5. **Pearson Correlation**: Measures the linear correlation between two variables.

$$r = \frac{\sum (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum (A_i - \bar{A})^2 \sum (B_i - \bar{B})^2}}$$

---

8. Why Normalized Versions of the Simple Matching Coefficient are Used for Measures of Association

Normalized versions of the **simple matching coefficient** are used to account for differences in the scale or dimensionality of data, ensuring that similarity measures are more accurate, especially when the variables involved have different ranges or distributions. This prevents bias toward longer or larger documents in information retrieval.

ASSIGNMENT-2

1. What is Clustering?

Clustering is an unsupervised machine learning technique that groups a set of objects (e.g., documents or data points) into clusters. Each cluster contains objects that are more similar to each other than to objects in other clusters. It's commonly used in information retrieval (IR) to organize similar items together.

---

2. Types of Clustering

1. **Hierarchical Clustering**:
    - Divides data into a hierarchy of clusters.
    - **Types**: Agglomerative (bottom-up), Divisive (top-down).
2. **Partitional Clustering**:
    - Divides data into distinct, non-overlapping clusters.
    - **Example**: K-means.
3. **Density-Based Clustering**:
    - Forms clusters based on data density (regions of high density separated by regions of low density).
    - **Example**: DBSCAN.
4. **Model-Based Clustering**:
    - Assumes data is generated by a mixture of probability distributions.
    - **Example**: Gaussian Mixture Models (GMM).
5. **Grid-Based Clustering**:
    - Divides the data space into grids and then clusters the objects based on the grids.
    - **Example**: STING (Statistical Information Grid).

---

3. Explain Single Pass Clustering Algorithm

The **Single Pass Clustering Algorithm** is a simple clustering method where each document or data point is processed one by one, and either assigned to an existing cluster or used to create a new one.

Steps:

1. Start with the first data point, and create the first cluster.
2. For each new data point, compute its similarity to the existing clusters.
3. If the similarity exceeds a pre-set threshold, assign the point to the closest cluster.
4. If not, create a new cluster for that point.

---

4. Explain Cluster Using Similarity Measures

Clusters can be formed based on **similarity measures** that quantify how close or similar two data points (documents) are. Common similarity measures include:

- **Cosine Similarity**: Measures the cosine of the angle between two vectors (often used in text retrieval).
- **Euclidean Distance**: Measures the straight-line distance between two points in space.
- **Jaccard Index**: Compares the similarity between two sets (e.g., sets of words in documents).

Clustering methods like K-means or hierarchical clustering use these similarity measures to group documents.

---

5. IR Models (Information Retrieval Models)

1. **Boolean Model**:

    - Uses logical operators (AND, OR, NOT) to match documents with queries.
2. **Vector Space Model (VSM)**:
    - Represents documents and queries as vectors in multi-dimensional space, using cosine similarity to rank results.
3. **Probabilistic Model**:
    - Ranks documents based on the probability that they are relevant to a given query.
4. **Language Models**:
    - Uses statistical models of language to rank documents by how likely they are to generate the query.
5. **Neural Network-based Models**:
    - Utilizes deep learning techniques for document and query understanding.

---

6. Boolean Search

Boolean search is a technique that uses Boolean logic (AND, OR, NOT) to combine search terms, allowing users to refine search queries by controlling which terms must or must not appear in the results.

- **AND**: Narrows the search by requiring all specified terms.
- **OR**: Expands the search by including results with any of the specified terms.
- **NOT**: Excludes results containing a specific term.

---

7. What is the Multi-pass Clustering Technique?

Multi-pass Clustering is an iterative extension of the single pass clustering algorithm. Instead of clustering data in a single pass, it refines clusters through multiple passes. After each pass, clusters are reviewed and adjusted, allowing for more accurate grouping of objects.

---

8. Explain Clustering Using Dis-Similarity Matrix. Effect of Threshold on Clustering

In clustering using a **dissimilarity matrix**, objects are represented as pairwise distances or dissimilarities. The matrix is used to guide the clustering process (e.g., hierarchical clustering).

Effect of Threshold on Clustering:

- **Lower Threshold**: Allows more objects to be included in clusters, resulting in fewer, larger clusters.
- **Higher Threshold**: Creates more distinct, smaller clusters, as the similarity requirement is stricter.

---

9. Explain K-list

A **K-list** is a data structure used in information retrieval and clustering to keep track of the **top K** most similar items for each data point. It is commonly used in the context of clustering algorithms to find the nearest neighbors or closest clusters.

---

10. Explain Cluster-Based Retrieval

Cluster-Based Retrieval is an IR technique where documents are pre-clustered, and when a query is submitted, instead of searching the entire document collection, the system first identifies relevant clusters and only searches within them. This improves search efficiency and relevance by focusing on similar documents.

ASSIGNMENT-3

1. What are Inverted Files?

Inverted files (or inverted indexes) are data structures used in information retrieval systems, such as search engines, to store a mapping from content (like words) to the locations (documents) where they occur. They allow for efficient full-text searches by linking each term to a list of documents where it appears.

---

2. What is Indexing?

Indexing is the process of creating a data structure (such as an inverted index) to make searching for information in large datasets efficient. It involves extracting and organizing relevant information from documents so that queries can retrieve the required results quickly.

---

3. What is Vocabulary and Occurrences?

- **Vocabulary**: The set of all distinct terms (words or tokens) in a collection of documents.
- **Occurrences**: The positions or counts where specific terms from the vocabulary appear in each document.

---

4. How Search is Carried Out on Inverted Index?

1. **Query Input**: A user submits a query with keywords.
2. **Term Lookup**: The search system looks up each term in the inverted index to find the list of documents (postings) where the terms appear.
3. **Intersection/Ranking**: For multiple keywords, the system finds documents containing all the terms (AND operation) or any terms (OR operation) and ranks them based on relevance (e.g., frequency, proximity).
4. **Results Retrieval**: The ranked results are returned to the user.

---

5. How to Index Multimedia Objects?

To index **multimedia objects** (images, audio, video), content-based features are extracted instead of text. Features such as color histograms (for images), audio frequency patterns, or motion vectors (for video) are indexed to allow for similarity-based search.

---

6. Limitations of Inverted Index

1. **High Storage Requirement**: For large datasets, inverted indexes can require significant storage space due to the need to store term-document mappings.
2. **Difficulty Handling Complex Queries**: Inverted indexes are optimized for simple keyword searches but may struggle with semantic or natural language queries.
3. **Inefficient for Real-Time Updates**: When documents are frequently added or modified, maintaining the inverted index efficiently can be challenging.
4. **Limited Multimedia Support**: It works well for text, but is less effective for multimedia objects without specialized indexing techniques.

---

7. What is Suffix-Array and Suffix-Tree?

- **Suffix Array**: A data structure that holds the sorted order of all suffixes of a string. It's useful in pattern matching and text retrieval.
- **Suffix Tree**: A compressed trie that represents all suffixes of a string. It allows for efficient substring searching and can answer queries related to pattern matching and string analysis in linear time.

---

8. What is the Concept of Signature Files?

Signature files are an indexing technique where each document is represented by a **fixed-length binary signature** (bit vector) created by hashing terms in the document. When searching, the query is hashed, and documents whose signatures match the query's signature are considered as potential matches.

---

9. Working of Inverted Files

1. **Document Processing**: Each document is processed, and terms are extracted.
2. **Index Creation**: For each term, a list of documents (postings list) where the term occurs is created.
3. **Query Execution**: When a query is submitted, the search engine checks the inverted file, retrieves the list of documents containing the search terms, and ranks them.
4. **Result Presentation**: The most relevant documents are returned to the user based on ranking criteria.

---

10. Applications of Inverted Index

1. **Search Engines**: Used in web search engines to enable fast text-based search.
2. **Database Systems**: For full-text indexing in database management systems.
3. **Information Retrieval**: In digital libraries, document management systems, and legal document searches.
4. **Plagiarism Detection**: Inverted indexes can identify copied or reused content by efficiently locating matching text.

---

11. Working of Signature Files

1. **Signature Creation**: Each document is converted into a signature (bit vector) using a hashing mechanism.
2. **Query Signature**: The query terms are hashed into a query signature.
3. **Comparison**: The query signature is compared to the document signatures, filtering out non-matching documents.
4. **Post-Filtering**: Documents that match the signature are retrieved and undergo further matching to ensure relevance.

ASSIGNMENT-4

1. What is Precision and Recall in IR System?

- **Precision**: The fraction of retrieved documents that are relevant.

  Precision=Relevant Documents RetrievedTotal Documents Retrieved\text{Precision} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Documents Retrieved}}Precision=Total Documents RetrievedRelevant Documents Retrieved

- **Recall**: The fraction of relevant documents that are retrieved.

  Recall=Relevant Documents RetrievedTotal Relevant Documents in the Collection\text{Recall} = \frac{\text{Relevant Documents Retrieved}}{\text{Total Relevant Documents in the Collection}}Recall=Total Relevant Documents in the CollectionRelevant Documents Retrieved

---

2. What is Relevance of a Document?

Relevance of a document refers to how well the document satisfies the user's information need or how closely it matches the user's query. In IR systems, relevance is often subjective and based on factors such as topic matching, content quality, and context.

---

3. What are the Metrics to Measure Information Systems?

1. **Precision**
2. **Recall**
3. **F-measure** (harmonic mean of precision and recall)
4. **E-measure**
5. **Mean Average Precision (MAP)**
6. **Discounted Cumulative Gain (DCG)**
7. **Normalized Discounted Cumulative Gain (NDCG)**
8. **Fall-out**: Measures the fraction of non-relevant documents retrieved.

---

4. How are Precision and Recall Calculated for Information Systems (Formulae)?

- **Precision**:

  Precision=True PositivesTrue Positives+False Positives\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}Precision=True Positives+False PositivesTrue Positives

- **Recall**:

  Recall=True PositivesTrue Positives+False Negatives\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}Recall=True Positives+False NegativesTrue Positives

---

5. What is the Problem with These Two Measures?

The main problem with **precision** and **recall** is that they often conflict with each other:

- Increasing **precision** typically decreases **recall**, as retrieving fewer documents can improve the relevance ratio but miss many relevant ones.

- Increasing **recall** often lowers **precision**, as retrieving more documents may result in more irrelevant ones being included.

---

6. What is Precision-Recall Trade-off?

The **Precision-Recall Trade-off** refers to the balance between achieving high precision and high recall. In an IR system:

- High precision means fewer irrelevant documents, but possibly missing relevant ones.
- High recall means retrieving more relevant documents, but at the cost of retrieving irrelevant ones too. The trade-off is finding the best balance between retrieving all relevant documents (recall) and minimizing irrelevant ones (precision).

ASSIGNMENT-5

1. **What is Harmonic Mean (F-measure) and E-measure in IR Systems?**

- **Harmonic Mean (F-measure)**: A measure that combines precision and recall into a single score by calculating the harmonic mean of the two. It gives more balanced weighting when precision and recall are at odds.

  Fβ=(1+β2)·Precision·Recallβ2·Precision+RecallF_\beta = \frac{(1 + \beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}Fβ=β2·Precision+Recall(1+β2)·Precision·Recall

  where **β** is a parameter that weights recall more if β > 1 and precision more if β < 1. The most common version is $\mathbf{F_1}$, where β = 1.

  F1=2·Precision·RecallPrecision+RecallF_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}F1=2·Precision+RecallPrecision·Recall

- **E-measure**: Combines precision and recall with a different weighting. It provides more flexible control over the relative importance of precision and recall by including a weighting parameter (α).

  Eα=1−(1α·1Precision+(1−α)·1Recall)E_\alpha = 1 - \left(\frac{1}{\alpha \cdot \frac{1}{\text{Precision}} + (1-\alpha) \cdot \frac{1}{\text{Recall}}} \right)Eα=1−(α·Precision1+(1−α)·Recall11)

  where **α** is a weight that adjusts the importance of precision and recall ($0 \leq \alpha \leq 1$).

---

2. How are F-measure and E-measure Calculated (Formulae)?

- **F-measure ($\mathbf{F_1}$)**:

  F1=2·Precision·RecallPrecision+RecallF_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}F1=2·Precision+RecallPrecision·Recall

- **E-measure**:

  Eα=1−(1α·1Precision+(1−α)·1Recall)E_\alpha = 1 - \left(\frac{1}{\alpha \cdot \frac{1}{\text{Precision}} + (1-\alpha) \cdot \frac{1}{\text{Recall}}} \right)Eα=1−(α·Precision1+(1−α)·Recall11)

---

3. What is the Difference Between F-measure and E-measure?

- **F-measure** is a harmonic mean that balances precision and recall equally when β = 1.
- **E-measure** is more flexible, allowing you to adjust the importance of precision versus recall using a weighting factor (α).

The key difference is the weighting system: F-measure uses a fixed harmonic mean, while E-measure allows for customizable weighting between precision and recall.

---

4. What are the Metrics to Measure Information Systems?

- **Precision**
- **Recall**
- **F-measure**
- **E-measure**

- **Mean Average Precision (MAP)**
- **Discounted Cumulative Gain (DCG)**
- **Fall-out**

---

5. What is the Advantage of F-measure and E-measure?

- **F-measure**: Balances precision and recall into a single metric, making it useful for evaluating systems where both precision and recall are important.
- **E-measure**: Offers more flexibility by allowing the adjustment of the importance of precision relative to recall through the α parameter, making it more customizable for different applications.

ASSIGNMENT-6

1. What is Extraction (or Feature Extraction)?

Feature extraction is the process of identifying and extracting meaningful information or attributes (features) from raw data (such as text, images, or audio). The goal is to reduce the dimensionality of the data while retaining important information that is useful for tasks like classification, clustering, or information retrieval.

---

2. How Are Images Indexed?

Image indexing involves extracting features from an image (e.g., color, texture, shape) and storing these features in a structured way to facilitate efficient searching and retrieval. Common methods for indexing images include:

- **Color histograms**
- **Texture patterns** (e.g., Gabor filters)
- **Shape descriptors** (e.g., contours or edges)
- **Local features** (e.g., SIFT, SURF)

Once features are extracted, they are used to create indexes that can be quickly searched when users submit a query (e.g., for similar images).

---

3. Explain How Color Is Extracted from an Image

Color can be extracted from an image using a **color histogram**, which counts the number of pixels that have each possible color value in the image. Steps involved:

1. **Convert Image**: The image is converted into a specific color space (e.g., RGB, HSV).
2. **Quantize Colors**: The color space is divided into bins to reduce the number of unique colors.
3. **Count Pixel Frequencies**: The number of pixels falling into each bin (color range) is counted, creating a histogram that represents the color distribution of the image.

---

4. What Is Multimedia IR? Discuss Steps on Which Data Retrieval Relies

Multimedia Information Retrieval (Multimedia IR) is the process of searching and retrieving multimedia content (such as images, audio, video) based on the features of that content rather than text-based metadata.

Steps in Multimedia IR:

1. **Feature Extraction**: Extract features (color, texture, shape, etc.) from multimedia objects.
2. **Indexing**: Index these features for efficient search and retrieval.
3. **Querying**: Users submit queries, which could be keywords, images, or other forms.
4. **Matching**: The system compares the query against the indexed features using similarity measures.
5. **Ranking and Retrieval**: Results are ranked and presented to the user based on similarity or relevance.

---

5. What Is the Use of Image Features?

Image features (such as color, texture, shape, or edges) are used to represent the visual content of an image in a compact and descriptive form. These features are essential for:

- **Image classification** (identifying objects within images).
- **Image retrieval** (searching for images based on similarity).

- **Image recognition** (identifying specific objects or patterns).
- **Content-based image retrieval (CBIR)** systems.

---

6. Enlist Some of the Features of Image and Its Applications

Common Image Features:

1. **Color**: Represented by color histograms, often used in image retrieval.
    - **Applications**: Content-based image retrieval, object recognition.
2. **Texture**: Describes the surface properties of an image using patterns.
    - **Applications**: Medical image analysis, remote sensing.
3. **Shape**: Extracted using edge detection or contours to represent the geometry of objects.
    - **Applications**: Object detection, facial recognition.
4. **Local Features**: Points of interest in the image (e.g., SIFT, SURF).
    - **Applications**: Image matching, stitching.

---

7. How to Compare Two Images and Calculate the Relevancy?

To compare two images and calculate their **relevancy**, the following steps are generally taken:

1. **Feature Extraction**: Extract features (e.g., color histograms, textures) from both images.
2. **Similarity Measure**: Use a similarity measure such as:
    - **Euclidean Distance**: Measures pixel-wise distance between two images.
    - **Cosine Similarity**: Measures the cosine of the angle between feature vectors.
    - **Histogram Intersection**: Compares color histograms.
3. **Thresholding**: A threshold is applied to the similarity score to determine if the images are relevant.

---

8. Applications of Feature Extraction

1. **Image Classification**: Categorizing images based on extracted features.
2. **Facial Recognition**: Identifying people based on facial feature extraction.
3. **Medical Image Analysis**: Detecting anomalies or classifying medical scans.
4. **Content-Based Image Retrieval (CBIR)**: Searching for images based on visual features.
5. **Object Detection**: Identifying and locating objects within images or videos.
6. **Speech Recognition**: Extracting audio features for recognizing spoken words.
7. **Natural Language Processing (NLP)**: Extracting text features for sentiment analysis, topic modeling.

ASSIGNMENT-7

1. What Are Search Engines? Name Few of Them

A **search engine** is a software system designed to search for information on the internet. It retrieves and ranks results based on a user's query by scanning web pages indexed in its database.

Examples of Search Engines:

- Google
- Bing
- Yahoo
- DuckDuckGo
- Baidu

---

2. How Search Engine Works?

Search engines work through three primary processes:

1. **Crawling**: Web crawlers (bots) visit and scan web pages to gather information.
2. **Indexing**: The gathered information is stored and organized in a massive database.
3. **Ranking and Retrieval**: When a user enters a query, the search engine ranks the indexed pages based on relevance and retrieves the most relevant ones for display.

---

3. What is Web Crawling?

Web Crawling is the process where a search engine's bots (also known as crawlers or spiders) automatically navigate the internet to discover and scan web pages, gathering content (text, images, links, etc.) for indexing purposes.

---

4. What is Robot Exclusion Protocol (robot.txt)?

The **Robots Exclusion Protocol (robots.txt)** is a file placed in the root directory of a website. It tells search engine crawlers which pages or sections of the website should not be crawled or indexed.

---

5. What is the Significance of robots.txt?

The **significance of robots.txt** is to control how search engines interact with a website:

- **Prevent Crawling**: Prevent search engines from crawling sensitive or irrelevant parts of the website (e.g., admin pages, duplicate content).
- **Optimize Crawling Efficiency**: Focus crawlers on important pages and reduce server load by excluding non-essential areas.

---

6. What Are the Strategies Used by Crawler?

Strategies used by web crawlers:

1. **Breadth-First Search (BFS)**: Crawlers explore all the pages linked from the current page before moving to a deeper level.
2. **Depth-First Search (DFS)**: Crawlers dive deep into a page's links before coming back up to the higher levels.
3. **Priority-Based Crawling**: Crawlers prioritize pages based on factors like PageRank, update frequency, or importance.
4. **Focused Crawling**: Crawlers focus on specific topics or types of content rather than crawling the entire web.

---

7. What is PageRank?

PageRank is an algorithm developed by Google to rank web pages based on their importance. It assigns a score to each page based on the number and quality of links pointing to it. The more high-quality links a page has, the higher its rank.

---

8. What is the Significance of Dampening Factor?

The **dampening factor** (typically set to 0.85) is used in the **PageRank algorithm** to ensure that the importance score does not get stuck in infinite loops of linked pages. It simulates the probability that a user randomly clicks on a link on a page rather than continuing to click on links indefinitely.

Formula for PageRank with damping factor:

$$PR(A) = (1 - d) + d \times \left(\frac{PR(B_1)}{L(B_1)} + \frac{PR(B_2)}{L(B_2)} + ... \right)$$

where:

- **PR(A)**: PageRank of page A
- **d**: Dampening factor (typically 0.85)
- **PR(B_1), PR(B_2)**: PageRank of pages linking to A
- **L(B_1), L(B_2)**: Number of outbound links from pages B1, B2, etc.

---

9. What Are the Crawler Architectures?

Common **crawler architectures**:

1. **Centralized Architecture**: One central crawler collects and indexes data.
2. **Distributed Architecture**: Multiple crawlers working in parallel to cover different sections of the web.
3. **Peer-to-Peer Crawling**: Crawlers working in a decentralized manner, sharing responsibilities and data without a central authority.

---

10. Explain Harvest Architecture

Harvest Architecture is a distributed system designed for scalable internet resource discovery. It consists of:

1. **Gatherers**: Collect and filter data from web pages.
2. **Brokers**: Store the collected data, provide indexing, and facilitate search across multiple gatherers.
3. **Replicators**: Handle redundancy and replication of data to ensure high availability.

---

11. Explain the Working of Google Crawler

The **Google Crawler**, also known as **Googlebot**, works as follows:

1. **Crawling**: Googlebot starts by crawling pages from a list of known URLs (from sitemaps or previously discovered URLs).
2. **URL Discovery**: As it crawls, it discovers new pages by following links from known pages.
3. **Fetching**: It fetches the HTML, CSS, JavaScript, and images from the web pages.
4. **Indexing**: The content is analyzed, indexed, and stored in Google's database.
5. **Updates**: Googlebot continuously revisits pages to check for updates or changes.

---

12. Explain Challenges Involved in Searching Web

Challenges in Web Search:

1. **Scale**: The web is massive and constantly growing, making it hard to index every page.
2. **Dynamic Content**: Many web pages are generated dynamically or change frequently, making it difficult to keep the index updated.
3. **Spam and Irrelevant Content**: Search engines must filter out low-quality or spammy content.
4. **Diversity of Content**: The web contains multimedia, structured, and unstructured data, all of which need different approaches for indexing and retrieval.
5. **Personalization**: Users expect personalized search results, adding complexity to ranking algorithms.
6. **Data Privacy**: Search engines must respect user privacy, especially with increased regulations around data protection (e.g., GDPR).

ASSIGNMENT-8

1. What are APIs and Their Use?

API (Application Programming Interface) is a set of rules and protocols that allow different software applications to communicate with each other. APIs define the methods and data formats that applications can use to request services from other software systems.

Uses of APIs:

- **Data Access**: Retrieve data from external sources (e.g., weather data, financial data).
- **Interoperability**: Enable different systems to work together (e.g., connecting payment gateways).
- **Automation**: Automate processes by allowing programs to interact with each other.
- **Integration**: Allow the integration of third-party services into applications (e.g., Google Maps, social media).

---

2. How to Use API?

To use an API, follow these steps:

1. **Obtain API Key**: Most APIs require an API key, which is used for authentication.
2. **Formulate Request**: Make an HTTP request (GET, POST, etc.) to the API endpoint using the specified URL.
3. **Include Parameters**: Attach the necessary parameters (query string, headers) to the request.
4. **Process Response**: The API will return a response, usually in JSON or XML format, which can be processed and used in your application.

---

3. Which API Have You Used in Your Assignment-8? (We Have Used OpenWeatherMap API)

In **Assignment-8**, we used the **OpenWeatherMap API**, which provides weather data for locations around the world. It allows us to retrieve current weather, forecasts, and historical weather data by city, geographic coordinates, or zip code.

---

4. Explain API Used in Assignment 8:

API_URL = https://api.weatherapi.com/v1/current.json?key=0ffbc5c35b604366adb42044240210&q=

The API used in **Assignment-8** is from **WeatherAPI.com**, and it provides the current weather data.

- **API URL**: The endpoint used to make requests to get current weather data.
- **key**: The parameter used to authenticate the API request. In this case, the key is "0ffbc5c35b604366adb42044240210".
- **q**: The query parameter used to specify the location for which we want the weather data (can be a city name, coordinates, or postal code).

Example Request:

https://api.weatherapi.com/v1/current.json?key=0ffbc5c35b604366adb42044240210&q=London

ASSIGNMENT-9

1. What is Case Study?

A **case study** is an in-depth analysis of a real-life scenario or example used to illustrate principles, problems, or techniques. It involves detailed investigation and documentation of the problem and how it was solved, providing insights into its broader applications.

---

2. On Which Topic Have You Done Case Study?

In **Assignment-8**, the case study was conducted on **weather data retrieval using OpenWeatherMap API**, focusing on how APIs can be used to retrieve, process, and display real-time weather information.

---

3. What Are Recommendation Systems?

Recommendation Systems are algorithms that suggest relevant items (such as products, services, or content) to users based on their preferences, past behavior, or the behavior of similar users. They are widely used in e-commerce, media streaming, and social platforms.

---

4. How Are Recommendation Systems Classified (or Types)?

Recommendation Systems are commonly classified into two main types:

1. **Collaborative Filtering**: Recommends items based on the preferences and behaviors of other users.
2. **Content-Based Filtering**: Recommends items based on the attributes of the items and user preferences.

Other types include:

- **Hybrid Systems**: Combine collaborative filtering and content-based filtering.
- **Knowledge-Based Systems**: Use explicit knowledge about users and items for recommendations.

---

5. Explain Collaborative Filtering Recommendation of Documents and Products

Collaborative Filtering recommends items by leveraging the preferences and behaviors of other users. There are two types:

1. **User-Based Collaborative Filtering**: Recommends items that similar users have liked. For example, if User A and User B have similar movie preferences, the system recommends movies that User B has liked to User A.
2. **Item-Based Collaborative Filtering**: Recommends items that are similar to those the user has liked. For example, if a user likes a particular product, the system recommends products that other users who liked the same product also purchased.

Application:

- Recommending documents or products based on the purchasing or viewing history of similar users.

---

6. Explain Content-Based Filtering Recommendation of Documents and Products

Content-Based Filtering recommends items based on the characteristics of the items and the user's past preferences. This approach relies on the features of the items (e.g., keywords, genres) and matches them to the user's known preferences.

Steps:

1. **Item Representation**: Each item is represented by its attributes (e.g., genre, price, author).
2. **User Profile**: A profile is created based on the items the user has liked in the past.
3. **Matching**: Items that have similar attributes to those the user liked are recommended.

Application:

- Recommending documents or products that share similar features (e.g., recommending articles on similar topics).