Case Study Assignment-V:

Implement the Architectures of **Set-Theoretic IR Models:** **Fuzzy**, **Boolean Extended**, **Algebraic**: **Generalized Vector**, and **Latent Semantic Indexing**

Assignment Steps

- A step-by-step algorithm for a student to implement the architecture of a Set-Theoretic IR Model in a chosen application.
- **Note:** Each model will have unique steps based on its underlying theory. Follow the general structure while customizing it to the chosen IR model.
- **Step 1: Choose a Set-Theoretic IR Model and Application**
- Select a specific Set-Theoretic IR model: Fuzzy, Boolean Extended, Generalized Vector, or Latent Semantic Indexing.
- Choose an application where the selected IR model will be applied. Example applications include search engines, e-commerce sites, educational platforms, or content management systems.

Assignment Steps

- **Step 2: Define the Search Criteria**
- **Boolean Extended:** Define basic AND, OR, and NOT operations for querying documents.
- **Fuzzy:** Introduce fuzzy sets with degrees of membership for documents.
- **Generalized Vector:** Define how to represent documents and queries as vectors with term weighting schemes.
- **Latent Semantic Indexing (LSI):** Define a term-document matrix and plan dimensionality reduction techniques using Singular Value Decomposition (SVD).
- Specify how queries will be handled in the model and define how documents are indexed and searched within the chosen application.

Assignment Steps

- **Step 3: Create a Representation Scheme**
- **For Boolean Extended:** Identify how documents will be represented using a set of terms and how the extended Boolean model will handle query complexity.
- **For Fuzzy:** Assign fuzzy membership degrees to documents based on relevance to search terms.
- **For Generalized Vector:** Design vector space representation and term weighting.
- **For Latent Semantic Indexing:** Construct the term-document matrix and plan for dimensionality reduction.
- Create a mapping between user queries and document sets, ensuring relevance measures are incorporated.

Assignment Steps

- **Step 4: Implement Query Processing**
- **For Boolean Extended:** Build a query processor that can handle complex Boolean expressions using weighted terms.
- **For Fuzzy:** Implement fuzzy logic-based query processing where partial matches are supported, and degrees of relevance are calculated.
- **For Generalized Vector:** Implement cosine similarity measures for vector space models.
- **For LSI:** Implement SVD to reduce dimensionality, and create a method for comparing reduced space vectors for search relevance.

Assignment Steps

- **Step 5: Implement Ranking Algorithms**
- **For Boolean Extended:** Rank based on how well documents satisfy the Boolean expression, including partial satisfaction.
- **For Fuzzy:** Rank based on membership degrees of documents in the fuzzy sets.
- **For Generalized Vector:** Rank documents using cosine similarity scores between the query and document vectors.
- **For LSI:** Rank based on similarity in the reduced latent space.

- **Design the ranking system to prioritize documents most relevant to user queries.**

Assignment Steps

- **Step 6: Add Interactive Features**
- Enhance the user experience by incorporating interactive features such as:
- Query refinement tools (suggestions, filters).
- Document previews or highlighting of matched terms.
- **In Fuzzy systems**, allow users to adjust fuzziness thresholds.
- **In Boolean Extended**, provide visual feedback on query construction.

Assignment Steps

- **Step 7: Test Query Processing and Ranking**
- Run tests to validate the query processing and ranking functionalities. Check:
- **For Boolean Extended:** Verify that documents matching complex expressions are retrieved correctly.
- **For Fuzzy:** Ensure partial matches return appropriate relevance scores.
- **For LSI:** Test that the reduced latent space results in relevant document retrieval.

Assignment Steps

- **Step 8: Conduct User Testing**
- Engage real users or a test audience to search using the model. Collect feedback on:
- Search result relevance.
- Ease of query formulation and interpretation of results.
- The effectiveness of interactive features.
- Performance across various query types (broad, specific, ambiguous).

- **Use this feedback to refine the system and improve the user experience.**

Assignment Steps

- **Step 9: Create Documentation**
- Explain the chosen Set-Theoretic IR Model (e.g., Boolean, Fuzzy, Algebraic, etc.) and how it is implemented in the application.
- Provide instructions for users on how to utilize the search functionalities effectively.
- Include technical details about the underlying search algorithm and ranking system.

Assignment Steps

- **Step 10: Deploy and Gather Continuous Feedback**
- Deploy the application with the implemented Set-Theoretic IR Model.
- Regularly gather user feedback and monitor the system’s performance.
- Make iterative improvements to both the search functionalities and user interface based on this feedback.

Any
Questions
?