Assignment Project: Implementing Interference and Belief Network Models in Python

Objective:

Students will implement two probabilistic models—Interference Model and Belief Network—from scratch in Python. The goal is to understand the underlying principles of these models and their application in Information Retrieval (IR).

Project Overview

Part 1: Interference Model

Students will create a simplified version of the Interference Model to evaluate document relevance based on queries.

Part 2: Belief Network

Students will implement a basic belief network to represent the probabilistic relationships among variables in a small IR context.

Project Requirements

1. Understanding the Models:

Students must research and understand the mathematical foundations of both the Interference Model and the Belief Network.

2. Data Preparation:

Students will create a small dataset that includes:

- A set of documents (text strings).

- A set of queries.

- Relevance judgments (indicating whether a document is relevant to a query).

3. Implementation:

- Interference Model:

- Define the probabilities for query and document pairs.

- Implement a function that computes the probability of relevance based on these pairs.

- Write a retrieval function that ranks documents for a given query based on their computed relevance probabilities.

-Belief Network:

- Define a small network of variables (e.g., query, document relevance, document features).

- Implement functions to calculate joint and marginal probabilities.

- Create a function that computes the probability of document relevance given a query using Bayes' theorem.

4. Testing:

- Test both models using the created dataset.

- Evaluate and compare the relevance rankings produced by both models.

5. Documentation:

- Write a report that includes:

- A brief introduction to both models.

- Code explanations and flowcharts where applicable.

- Results of the model evaluations, including any observed differences in performance.

Step-by-Step Instructions

Part 1: Interference Model

1. Define the Dataset:

- Create a list of documents and queries.

- Create a dictionary to represent relevance judgments.

2. Compute Probabilities:

- Define functions to calculate probabilities for queries and documents based on the dataset.

3. Implement Retrieval Function:

- Create a function that returns a ranked list of documents for a given query based on their relevance probabilities.

Part 2: Belief Network

1. Define the Network Structure:

- Represent the relationships among variables using a simple structure.

2. Calculate Joint and Marginal Probabilities:

- Implement functions to calculate the probabilities necessary for the belief network.

3. Implement Bayes' Theorem:

- Create a function to compute (P(Relevance | Query)) using Bayes' theorem.

Testing and Evaluation

1. Test the Models:

- Run test queries against both models and collect the results.

2. Analyze Results:

- Compare the relevance rankings from both models and note differences.

Documentation

- Prepare a report summarizing:

- Model descriptions and implementations.

- Code snippets and explanations.

- Results and performance analysis.

- Reflections on the implementation process and challenges faced.

Submission Guidelines

- Submit the complete Python code in a single `.py` file on the GitHub.

- Include the documentation report in PDF format.

- Ensure code is well-commented and follows best practices for readability.

Grading Criteria

-Correctness: Accurate implementation of both models (40%).

-Code Quality: Clean, well-organized, and commented code (30%).

-Documentation: Clarity and completeness of the report (20%).

-Analysis: Insightful comparisons and reflections on the models (10%).

This project aims to provide hands-on experience with probabilistic models in information retrieval, enhancing both programming skills and theoretical understanding.