CS F441 Selected Topics in Computer Science [1st Semester 2024-2025]

Comprehensive Exam [16.12.2024] [Max Marks: 65] [Duration: 180 mins]

Build a conversational application where the LLM acts as a student and asks questions about a **single research paper on Generative AI Agents**. The human user (teacher) explains the paper's content. All dialogue must remain on-topic and grounded in the provided paper. You must implement a Retrieval-Augmented Generation (RAG) approach to derive question-answer pairs from the paper. Importantly, you are **not** allowed to feed the entire paper directly to the LLM. Instead, you must extract and process relevant sections to form a knowledge base from which the LLM can retrieve information.

Requirements:

1. **Data Preparation (RAG-Based):**
    - **Single Research Paper:**
      You are provided with one research paper in the domain of Generative AI Agents.
    - **Knowledge Extraction:**
      Use a RAG-based technique to extract relevant portions (e.g., motivation, problem statement, comparison with existing methods, proposed methodology, experiments, findings, and conclusions) from the paper.
    - **No Direct Full-Text Feed:**
      You cannot simply load the entire paper into the LLM's context.
    - **Conversation Generation:**
      The extracted portions will be used for generating conversations between the AI and the user as mentioned below.

2. **Conversation Design:**
    - Use LLMs to generate **5 multi-turn conversations** (4-5 turns each) between the AI (as a student) and the user (as a teacher) about the given research paper.
    - Each conversation should start with the AI asking a question related to a specific aspect of the paper.
    - **Example Types of AI Queries:**
        - **Conceptual Understanding:** "Can you explain how the generative model in the paper differs from traditional language models?"
        - **Methodology Details:** "I'm not sure I understand how the authors trained their agent. Could you clarify the training procedure?"
        - **Practical Application:** "How can the approach described in the paper be used to improve human-AI collaboration scenarios?"

- **Comparative Insight:** "The paper mentions related work. Can you tell me how this approach compares to previous methods mentioned?"
- **Clarification on Results:** "They talk about improved performance metrics. Could you explain what metrics they used and how the results improved?"
- **Follow-up Questions:** "Can you tell me more about the agent's reasoning process which you just mentioned?"

In each of the 5 conversations and across the 5 conversations, the AI's queries should be diverse, covering different question types. That is, there should be inter-conversation diversity and intra-conversation diversity.

3. **Guardrails (Using Nemo Guardrails):** Implement exactly two guardrails:

   i. **AI Query Validation Guardrail:**
   Before the AI's question is shown to the user, validate that it is grounded in the extracted data from the paper.

   - If the question is off-topic or unsupported by the research paper, trigger a regeneration step.
   - Try up to 2 times. If after 2 attempts, the AI still fails, display the last generated question as is.

   ii. **User Response Validation Guardrail:**
   Validate the user's response to ensure that they remain on-topic.

   - If the user's response includes content unrelated to the research paper, the AI should politely prompt the user to stay on topic. For example: "It seems you are talking about something not covered in the research paper. Could you please focus on the topics discussed in the research paper?"

4. **Evaluation (Using Ragas):** For simplicity, we will use the five conversations (generated in the data generation step) and evaluate both the user (teacher) responses and the AI (student) responses using the Ragas framework.

   User Response Evaluation Metrics (per conversation):

   i. **Fact-Checking (Claims Verification):**

   - Identify claims the user makes in their responses.
   - Verify each claim against the research paper content.
   - Compute the total number of correct claims and the total number of incorrect claims per conversation. Normalize the score by the total number of claims (correct + incorrect).

ii. **Explanatory Depth (Rubric-Based 1-5 Score):**

- Assess how well the user explains concepts.
- A score of 1 indicates very shallow, superficial explanations; 5 indicates deep, thorough, and context-rich explanations.
- Compute the average Explanatory Depth score for each conversation.

AI Response Evaluation Metrics (per conversation):

i. **Question Diversity:**

- Pre-define a set of question types (e.g., conceptual, methodological, practical application, comparative, clarification on results).
- Count how many distinct types of questions the AI asked within the conversation.
- Compute the normalized diversity: (Number of distinct question types) / (Number of AI turns in the conversation).

ii. **Paper Consistency:**

- Count the percentage of AI generated queries which are not grounded in the research paper.
- A higher count indicates lower consistency.

5. **Code Requirements:**

   - At a minimum, have three Python files:
     - `data_generation.py` : Implements RAG-based extraction from the single research paper and prepare a set of five conversations in `json` format.
     - `application.py` : Builds the conversational application which uses Nemo Guardrails for both AI query validation and user response validation.
     - `evaluation.py` : Uses Ragas framework to compute the specified evaluation metrics (fact-checking claims, explanatory depth for user; question diversity and paper consistency for AI) for each of the five conversations generated in the data generation step. You must not use any pre-defined Ragas metrics. You should implement your own Ragas metrics.

Important Notes:

- Submit three Python files as specified, all fully executable.
- Do not blindly generate the code from LLM. You will be asked to explain the code line by line during the answer sheet evaluation.