|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | |
| **Program Name:** B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | |
| **Instructor(s)Name** | | | | 1. Dr. Mohammed Ali Shaik  2. Dr. T Sampath Kumar  3. Mr. S Naresh Kumar  4. Dr. V. Rajesh  5. Dr. Brij Kishore  6. Dr Pramoda Patro  7. Dr. Venkataramana  8. Dr. Ravi Chander  9. Dr. Jagjeeth Singh | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | |
| **Date and Day**  **of Assignment** | | | 06-08-2025 | **Time(s)** | |  | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | |
| **AssignmentNumber:4.5**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | |
|  | | | | | | | | |
|  | **Q. No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | **Lab 4: Advanced Prompt Engineering: Zero-shot, one-shot, and few-shot techniques**  **Objective:** To explore and compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories using a large language model (LLM).  Suppose that you work for a company that receives hundreds of customer emails daily. Management wants to automatically classify emails into categories like "Billing", "Technical Support", "Feedback", and "Others" before assigning them to appropriate departments. Instead of training a new model, your task is to use prompt engineering techniques with an existing LLM to handle the classification.  Tasks to be completed are as below  1. **Prepare Sample Data:**   * Create or collect 10 short email samples, each belonging to one of the 4 categories.   Prompt: Generate 10 short emails regarding the given scenario.( Suppose that you work for a company that receives hundreds of customer emails daily. Management wants to automatically classify emails into categories like "Billing", "Technical Support", "Feedback", and "Others" before assigning them to appropriate departments. )  2. **Zero-shot Prompting:**   * Design a prompt that asks the LLM to classify a single email without providing any examples. * Example prompt: *“Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others. Email: ‘I have not received my invoice for last month.’”*   *Prompt : generate a python code which takes email input from user and categorize of which one it belongs to from the four categories. (: Billing, Technical Support, Feedback, Others) .*    **3. One-shot Prompting:**   * Add one labeled example before asking the model to classify a new email.   Prompt : generate a python code which takes email input from user and categorize of which one it belongs to from the four categories. (: Billing, Technical Support, Feedback, Others) .(( Example prompt: “Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others. Email: ‘I have not received my invoice for last month.’”))    4. **Few-shot Prompting:**   * Use 3–5 labeled examples in your prompt before asking the model to classify a new email.   Prompt:  generate a python code which takes email input from user and categorize of which one it belongs to from the four categories. (: Billing, Technical Support, Feedback, Others). generate this such that i can sought out 3 to 5 emails at a time      5. **Evaluation:**   * Run all three techniques on the same set of 5 test emails. * Compare and document the accuracy and clarity of responses.   Prompt :  Can u now check all the three techniques (zero shot, one shot, two shot) on the same set of 5 test emails and generate output.  **Explanation:**   * **Generate test emails: Create a list of 5 test emails that cover the different categories (Billing, Technical Support, Feedback, and Others).** * **Set up language model: Initialize the Gemini language model for classification.** * **Implement zero-shot classification: Classify the test emails using a zero-shot approach (providing no examples).** * **Implement one-shot classification: Classify the test emails using a one-shot approach (providing one example per category).** * **Implement two-shot classification: Classify the test emails using a two-shot approach (providing two examples per category).** * **Present results: Display the classification results for each technique alongside the original emails.** * **Finish task: Summarize the findings and compare the performance of each technique**   **Summary:**  **Data Analysis Key Findings**   * **The code successfully generated a list of 5 test emails covering the categories: Billing, Technical Support, Feedback, and Others.** * **The code successfully set up the Gemini Pro language model using the provided API key.** * **The implementation of zero-shot, one-shot, and two-shot classification techniques was attempted, but the execution consistently failed due to a DefaultCredentialsError. This error indicates that the Google API key was not accessible in the environment.** * **Due to the persistent API authentication error, the classification results for all techniques were not successfully generated and stored, leading to "N/A - Error in classification" being displayed during the results presentation step.**   **Insights or Next Steps**   * **The primary next step is to ensure the correct configuration of the Google API key in the execution environment to resolve the DefaultCredentialsError and enable successful interaction with the Gemini model for classification.** * **Once the API key issue is resolved, re-running the classification steps will provide the actual classification results for comparison across the zero-shot, one-shot, and two-shot techniques.**   **Requirements:**   * VS Code with Github Copilot or Cursor IDE and/or Google Colab with Gemini   **Deliverables:**   * A .txt or .md file showing prompts and model responses. * A comparison table showing classification accuracy for each technique. * A short reflection on which method was most effective and why   . | | | | | | 08.08.2025 EOD |  |