|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | | | ***Ex pe***  ***ct***  ***ed***  ***Ti m e to co m pl et e*** |
|  | **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). | | | | | | 08.0  8.20 |
| 1 | Suppose that you work for a company that receives hundreds of customer emails daily. Management wants | | | | | | 25 |
|  | to automatically classify emails into categories like "Billing", "Technical Support", "Feedback", and | | | | | | EO |
|  | "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 | | | | | | D |

|  |  |  |
| --- | --- | --- |
|  | 1. **Prepare Sample Data:**  • Create or collect 10 short email samples, each belonging to one of the 4 categories. **Prompt:**  1. Create 10 brief email examples, categorized into four groups: Billing, Technical Support, Feedback, and Miscellaneous.    **Code:** |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **Code Explanation**     1. A dictionary is created with 4 categories. 2. Each category has a list of short emails. 3. A loop goes through each category. 4. Another loop prints the emails inside that category. 5. The program shows the category name first, then its emails.     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**  Classify the given email into one of the following groups: Billing, Technical Support, Feedback, or Others  **Code:** |  |

|  |  |  |
| --- | --- | --- |
|  | **Code Explanation:**   1. The program asks the user to enter an email. 2. Some keywords are given for each category. 3. The program checks the email for those keywords. 4. If a keyword is found, it assigns that category.     **3. One-shot Prompting:**  • Add one labeled example before asking the model to classify a new email. **Prompt**  Sort the email into the appropriate category: **Billing, Technical Support, Feedback, Others.**  Example:  Email: *‘The app keeps crashing when I try to log in.’* → Category: **Technical Support** Now classify the user input:  Email: *‘I am unable to reset my password.’*”  **Code:** |  |

|  |  |  |
| --- | --- | --- |
|  | **Code**    **Explanation:**     1. A sample email is provided as a reference. 2. The model is then prompted to categorize the user’s email. 3. A mock classifier is used to simulate the output.       **4.. Few-shot Prompting:**  a. Use 3–5 labeled examples in your prompt before asking the model to classify a new email.  **Prompt**  Decide which category the email fits into: **Billing, Technical Support, Feedback, or Other**  Examples:  Email: *‘I cannot log into my account.’* → Category: Technical Support  Email: *‘I love the new features in your app.’* → Category: Feedback  Email: *‘I was charged twice for the same service.’* → Category: Billing Email: *‘Can you tell me your office hours?’* → Category: Others    Now classify the user input      **Code:** |  |

|  |  |  |
| --- | --- | --- |
|  | **Code Explanation**   1. A small set of labeled examples (3–5) is presented as reference. 2. The model applies this context to classify the unseen email. 3. A simulated classifier is used to display the predicted category.     5. **Evaluation:**   * Run all three techniques on the same set of 5 test emails. * Compare and document the accuracy and clarity of responses.   **Prompt**  Classify the following 5 emails into one of the categories: Billing, Technical Support, Feedback, Others.  Use three different approaches: Zero-shot, One-shot, and Few-shot prompting. Compare the outputs for each approach **Code:** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Code Explanation:**  1.5 test emails are defined.   1. Each email is classified using zero shot, one shot and few shot. 2. Results are stored in a table.       **Comparison Table**  **Email** | **Zero-shot** | **One-shot** | **Few-**  **shot** | | |  |
|  | | I want to update my credit card details for billing. | Billing | Billing | Billing | | |  |
|  | | The app keeps freezing on my phone. | Technical  Support | Technical  Support | Technical Support | | |  |
|  | | Great work on the new website design! | Feedback | Feedback | Feedback | | |  |
|  | | Can you share your holiday schedule? | Others | Others | Others | | |  |
|  |  | **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   . | | |  |  |  |
|  | |  | | |  | |