AI ASSISTED CODING

HALL TICKET:2403A51270

BATCH 12

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. 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 08.08.2025 EOD for last month.’” 3. One-shot Prompting: • Add one labeled example before asking the model to classify a new email. 4. Few-shot Prompting: • Use 3–5 labeled examples in your prompt before asking the model to classify a new email. 5. Evaluation: • Run all three techniques on the same set of 5 test emails. • Compare and document the accuracy and clarity of responses. 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

**1. 📬 Sample Email Data**

| **Email ID** | **Email Text** | **True Category** |
| --- | --- | --- |
| E1 | I haven't received my bill for last month. | Billing |
| E2 | The app crashes every time I try to upload a file. | Technical Support |
| E3 | Just wanted to say how impressed I am with your new feature! | Feedback |
| E4 | Can you help me update my credit card on file? | Billing |
| E5 | I'm not sure if this is the right place, but I have a general question. | Others |
| E6 | The technician was very helpful. Kudos to your team! | Feedback |
| E7 | My login credentials aren’t working anymore. | Technical Support |
| E8 | How do I cancel my subscription before next billing cycle? | Billing |
| E9 | I want to suggest a new feature: dark mode. | Feedback |
| E10 | I don't know where else to send this, so I'm emailing you here. | Others |

Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others.

Email: "The app crashes every time I try to upload a file."

**Expected Output:** Technical Support  
(Repeat for 5 test emails: E2, E3, E5, E7, E9)

### 1. ****Zero-shot Prompting****

**Definition:**  
You ask the model to do a task **without giving any examples**.

**Example Prompt:**

Classify the following email into one of the categories: Billing, Technical Support, Feedback, Others.

Email: "I can't log into my account."

**How it works:**  
The model uses its general knowledge to answer, even without seeing a similar example.

### 2. ****One-shot Prompting****

**Definition:**  
You give the model **one example** before asking it to complete a similar task.

Example:

Email: "I need help with my payment."

Category: Billing

Now classify this email:

Email: "I can't log into my account."

Category:

**How it works:**  
The model uses the single example to better understand the format and context.

### 3. ****Few-shot Prompting****

**Definition:**  
You give the model **a few examples** (usually 3–5) before asking it to do the task.

Examples:

Email: "I need help with my payment."

Category: Billing

Email: "The app isn't loading properly."

Category: Technical Support

Email: "Great job on the new design!"

Category: Feedback

Now classify this email:

Email: "I can't log into my account."

Category:

**How it works:**  
The model learns from the examples and can perform the task more accurately.

### Summary Table:

| **Technique** | **Number of Examples** | **Accuracy (Usually)** | **Use When...** |
| --- | --- | --- | --- |
| Zero-shot | 0 | Medium | Task is very clear or simple |
| One-shot | 1 | Better | Slightly complex task |
| Few-shot | 3–5 | Best | Task is complex or requires pattern learning |

## EVALUATION OF EMAIL CLASSIFICATION

You're comparing how well each prompting method performs when classifying the same set of test emails. Here's how to structure the evaluation:

### 1. ****Test Emails for Evaluation****

Use 5 emails (examples below):

| **Test ID** | **Email Text** | **True Category** |
| --- | --- | --- |
| T1 | "The app crashes every time I try to upload a file." | Technical Support |
| T2 | "Just wanted to say how impressed I am with your new feature!" | Feedback |
| T3 | "I'm not sure if this is the right place, but I have a general question." | Others |
| T4 | "My login credentials aren’t working anymore." | Technical Support |
| T5 | "I want to suggest a new feature: dark mode." | Feedback |

### 2. ****Run Each Prompt Type on These Emails****

For each test email, run:

* The **Zero-shot prompt**
* The **One-shot prompt**
* The **Few-shot prompt**

### 3. ****Create a Comparison Table****

Example:

| **Email ID** | **True Category** | **Zero-shot Prediction** | **One-shot Prediction** | **Few-shot Prediction** | **Zero Correct?** | **One Correct?** | **Few Correct?** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | Technical Support | Technical Support | Technical Support | Technical Support | ✅ | ✅ | ✅ |
| T2 | Feedback | Others | Feedback | Feedback | ❌ | ✅ | ✅ |
| T3 | Others | Others | Others | Others | ✅ | ✅ | ✅ |
| T4 | Technical Support | Feedback | Technical Support | Technical Support | ❌ | ✅ | ✅ |
| T5 | Feedback | Feedback | Feedback | Feedback | ✅ | ✅ | ✅ |

### 4. ****Calculate Accuracy****

For each method:

Accuracy = (Number of correct predictions / Total test emails) × 100

Example:

| **Technique** | **Accuracy** |
| --- | --- |
| Zero-shot | 3 / 5 = 60% |
| One-shot | 5 / 5 = 100% |
| Few-shot | 5 / 5 = 100% |
| Project Requirements🖥️ ****Environment Requirements**** You should use one of the following tools: Option 1: ****VS Code with GitHub Copilot or Cursor IDE****  * Visual Studio Code installed * GitHub Copilot extension enabled (to get AI-assisted prompts/responses) * Alternatively, use **Cursor IDE** (a Copilot-enhanced IDE)  Option 2: ****Google Colab with Gemini****  * A Google account to access Google Colab * Access to **Gemini** or other LLM API (like PaLM 2 or OpenAI via API key) * Internet connection to run and test prompts  ****Data & Prompt Requirements****  1. **Sample Email Dataset**    * 10 short, realistic customer emails    * Each email should belong to one of the 4 categories:      + **Billing**      + **Technical Support**      + **Feedback**      + **Others** 2. **Prompting Techniques**    * **Zero-shot prompting**: No example, only the task description    * **One-shot prompting**: One example + task    * **Few-shot prompting**: 3–5 examples + task 3. **Test Set**    * Select **5 emails** (from your dataset or new ones) for consistent testing  ****Evaluation Requirements****  * Classify all 5 test emails using **each prompting technique** * Compare predictions to **true categories** * Create a **comparison table** to show results * Calculate **accuracy (%)** for each technique * Write a **short reflection** on which method worked best and why  ****Deliverables****  | **Task** | **Status** | | --- | --- | | 🔲 Create 10 sample emails |  | | 🔲 Write 3 types of prompts |  | | 🔲 Run classification on 5 test emails |  | | 🔲 Record and compare results |  | | 🔲 Calculate accuracy |  | | 🔲 Write reflection |  | | 🔲 Submit .md or .txt file with all content |  |  | **Deliverable** | **Description** | | --- | --- | | ✅ prompts\_and\_responses.md or .txt | File containing your sample emails, prompts, and LLM responses | | ✅ Comparison Table | Table comparing results across Zero-shot, One-shot, Few-shot | | ✅ Accuracy Summary | How accurate each method was (e.g. 3/5 = 60%) | | ✅ Reflection | A short paragraph explaining which method worked best and why | | ✅ (Optional) Screenshots or Colab Link | If you're using Copilot or Gemini, include screenshots or a link showing your work | |  |