

# **AI ASSISTANT CODING**

## **ASSIGNMENT – 4.5**

**NAME: K.SHAILAJA**

**HALL TICKET NO.: 2303A52155**

**BATCH: 34**

### **QUESTION:**

Lab 4: Advanced Prompt Engineering: Zero-shot, one-shot, and few-shot techniques Week2 -

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).

1. 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

#### **a. Prepare Sample Data:**


- Create or collect 10 short email samples, each belonging to one of the 4 categories.

#### **Prompt:**

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. Analyse the given scenario and give me the 5 test emails

**Output:**

Test ID	Email Content	
Test_01	""I tried to upgrade to the Pro plan, but my credit card was declined three times. Can you help me finish the transaction?""	
Test_02	""The app crashed while I was in the middle of a project, and now I've lost three hours of work. This is unacceptable and very frustrating.""	
Test_03	""I really think adding a 'Export to PDF' button would make the reports much more useful for my weekly meetings.""	
Test_04	""Hi, I'm a tech blogger and I'd love to interview your CEO for an upcoming piece on industry innovators. Who should I contact?""	
Test_05	""I was overcharged by \$10, and when I tried to log in to fix it, the login page wouldn't load.""	

**b. 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:**

Act as a Email Classifier and classify the following email into one of the following categories:

Billing, Technical Support, Feedback, Others.

### Output:

Test ID	Email Content	Category	Reasoning	
Test_01	""I tried to upgrade to the Pro plan, but my credit card was declined three times. Can you help me finish the transaction?""	Billing	This involves a failed financial transaction and payment issue.	
Test_02	""The app crashed while I was in the middle of a project, and now I've lost three hours of work. This is unacceptable and very frustrating.""	Technical Support	This addresses a software functional failure (a crash).	
Test_03	""I really think adding a 'Export to PDF' button would make the reports much more useful for my weekly meetings.""	Feedback	This is a suggestion for a new feature or product improvement.	
Test_04	""Hi, I'm a tech blogger and I'd love to interview your CEO for an upcoming piece on industry innovators. Who should I contact?""	Others	This is a general media inquiry unrelated to support or billing.	
Test_05	""I was overcharged by \$10, and when I tried to log in to fix it, the login page wouldn't load.""	Technical Support	"While it mentions billing, the ""tie-breaker"" rule prioritizes the technical issue (inability to log in) to ensure the user can access the system first."	

### c. One-shot Prompting:

- Add one labeled example before asking the model to classify a new email.

#### Prompt:

Analyse the given example and Classify the following email into one of the following categories:

Billing, Technical Support, Feedback, based on their context on given image of emails

## Output:

Test ID	Email Content	Category	Reasoning from Images
Test_01	""I tried to upgrade to the Pro plan, but my credit card was declined three times. Can you help me finish the transaction?""	Billing	This prioritizes the financial transaction issue over the general request for help.
Test_02	""The app crashed while I was in the middle of a project, and now I've lost three hours of work. This is unacceptable and very frustrating.""	Technical Support	"The core issue is a technical failure (the crash), which takes precedence over the customer's frustration."
Test_03	""I really think adding a 'Export to PDF' button would make the reports much more useful for my weekly meetings.""	Feedback	This is identified as a feature request or suggestion rather than a technical problem.
Test_04	""Hi, I'm a tech blogger and I'd love to interview your CEO for an upcoming piece on industry innovators. Who should I contact?""	Others	This is a professional inquiry that does not qualify as a support ticket.
Test_05	""I was overcharged by \$10, and when I tried to log in to fix it, the login page wouldn't load.""	Technical Support	"This hybrid test uses a ""Tie-breaker"" rule where Technical Support is prioritized over Billing."

## d. Few-shot Prompting:

- Use 3–5 labeled examples in your prompt before asking the model to classify a new email.

### Prompt:

You are an AI assistant that classifies customer emails into one of the following categories:

Billing: Payment failures, charges, refunds, subscription issues

Technical Support: App crashes, login issues, bugs, system failures

Feedback: Feature requests, suggestions, product improvements

Others: Media inquiries, partnerships, general questions not related to support

Follow these rules:

Identify the core problem, not the emotional tone.


If multiple issues exist, apply a tie-breaker:

Technical Support > Billing > Feedback > Others

Output must be in tabular format with columns:

Test ID | Email Content | Category | Reasoning

### Output:

Test ID	Email Content	Category	Reasoning	
Test_01	I tried to upgrade to the Pro plan, but my credit card was declined three times. Can you help me finish the transaction?	Billing	This prioritizes a failed financial transaction.	
Test_02	The app crashed while I was in the middle of a project, and now I've lost three hours of work.	Technical Support	The core issue is a system crash.	
Test_03	I really think adding an Export to PDF button would make reports more useful.	Feedback	This is a feature suggestion.	
Test_04	I'm a tech blogger and would like to interview your CEO. Who should I contact?	Others	This is a media inquiry, not a support request.	
Test_05	I was overcharged, and when I tried to log in to fix it, the login page wouldn't load.	Technical Support	Technical access issues take priority over billing.	

### e. Evaluation:

- Run all three techniques on the same set of 5 test emails.
- Compare and document the accuracy and clarity of responses.

### Output:

**Resolves Ambiguity:** It enables the model to apply "Tie-breaker" rules in hybrid cases, such as prioritizing Technical Support over Billing when both issues are present.

**Contextual Depth:** The model learns to prioritize core technical failures over customer sentiment, preventing it from misclassifying urgent bugs as mere Feedback.

**Boundary Precision:** By analyzing multiple examples, the model accurately distinguishes between similar categories, such as "Feature Requests" (Feedback) versus "Bugs" (Technical Support), far more effectively than guessing based on names alone.

## 2. Travel Query Classification

### Scenario:

A travel assistant must classify queries into Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

Tasks:

- Prepare labeled travel queries.**

**Prompt:**

These are the test emails we will use for the AI:

- Email 1: "I want to book a flight to Paris for next Monday."
- Email 2: "I need a hotel room for two people in Dubai."
- Email 3: "Can I cancel my trip and get a refund?"
- Email 4: "What is the weather like in London in May?"
- Email 5: "I want to change my hotel stay to a different date."

**Output:**

ID	Email Content	Category	Zero-shot Result	One-shot Result	Few-shot Result
1	"Reserve a round-trip ticket..."	<b>Flight Booking</b>	Flight Booking	Flight Booking	Flight Booking
2	"Cancel my reservation..."	<b>Cancellation</b>	Cancellation	Cancellation	Cancellation
3	"Double rooms available..."	<b>Hotel Booking</b>	Hotel Booking	Hotel Booking	Hotel Booking
4	"Is a visa required..."	<b>General Travel Info</b>	General Travel Info	General Travel Info	General Travel Info
5	"Change my flight date..."	<b>Cancellation</b>	Flight Booking	Cancellation	<b>Cancellation</b>

**b. Apply Zero-shot prompting.****Prompt:**

Classify this email into one category: Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

**Output:**

ID	Email content	Category	One-shot Result
1	"I want to book a flight to Paris for next Monday."	<b>Flight Booking</b>	Flight Booking
2	"I need a hotel room for two people in Dubai."	<b>Hotel Booking</b>	Hotel Booking
3	"Can I cancel my trip and get a refund?"	<b>Cancellation</b>	Cancellation
4	"What is the weather like in London in May?"	<b>General Travel Info</b>	General Travel Info
5	"I want to change my hotel stay to a different date."	<b>Cancellation</b>	Cancellation

### c. Apply One-shot prompting.

#### Prompt:

Classify this email into one category: Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

Example: Email content: "I need to book a ticket to Tokyo." Answer: Flight Booking

Task: Email content: [Insert Email Here] Answer:

#### Output:

ID	Email content	Category	One-shot Result
1	"I want to book a flight to Paris for next Monday."	<b>Flight Booking</b>	Flight Booking
2	"I need a hotel room for two people in Dubai."	<b>Hotel Booking</b>	Hotel Booking
3	"Can I cancel my trip and get a refund?"	<b>Cancellation</b>	Cancellation
4	"What is the weather like in London in May?"	<b>General Travel Info</b>	General Travel Info
5	"I want to change my hotel stay to a different date."	<b>Cancellation</b>	Cancellation

### d. Apply Few-shot prompting.

#### Output:

ID	Email content	Category	Few-shot Result
1	"I want to book a flight to Paris for next Monday."	<b>Flight Booking</b>	Flight Booking
2	:I need a hotel room for two people in Dubai."	<b>Hotel Booking</b>	Hotel Booking
3	"Can I cancel my trip and get a refund?"	<b>Cancellation</b>	Cancellation
4	"What is the weather like in London in May?"	<b>General Travel Info</b>	General Travel Info
5	"I want to change my hotel stay to a different date."	<b>Cancellation</b>	<b>Cancellation</b>

### e. Compare response consistency.

#### Output:

The results show that as we provide more examples, the AI becomes more consistent and accurate.

- Zero-shot is inconsistent because the AI has to guess the format, often leading to long or slightly incorrect answers.
- One-shot improves the format but can still struggle with "tricky" queries that don't match the single example provided.

- Few-shot provides the best consistency. By seeing a variety of queries, the AI clearly understands the boundaries between categories (like the difference between asking about a hotel and actually booking one).

### 3. Programming Question Type Identification

#### Scenario:

A coding help chatbot must classify queries into Syntax Error, Logic Error, Optimization, or Conceptual Question.


Tasks:

#### a. Prepare coding-related user queries.

#### Prompt:

- a. coding help chatbot must classify queries into Syntax Error, Logic Error, Optimization, or Conceptual Question.
- b. Error, Optimization, or Conceptual Question.
- c. Tasks:
- d. a. Prepare coding-related user queries which is adaptable to the given context

#### Output:

Category	Strategic Purpose	
Syntax Error	Tests the model's ability to identify explicit code-breaking errors that prevent execution.	
Logic Error	"Tests the model's ability to identify ""working"" code that produces incorrect outputs."	
Optimization	Tests the focus on resource efficiency (time/memory) rather than correctness.	
Conceptual Question	Tests for theoretical or general knowledge requests that do not involve specific code failures.	
Hybrid (Test Case)	The Tie-Breaker Test: Designed to see if the model prioritizes the Syntax Error (blocker) over the Optimization request.	

#### b. Perform Zero-shot classification.

#### Prompt:

Classify the given query into one category (Syntax Error, Logic Error, Optimization, or Conceptual Question) by identifying the core issue and applying the priority order Syntax Error > Logic Error > Optimization > Conceptual Question, then output the Category

#### Output:



Query 1: "I'm getting IndentationError: expected an indented block on line 12 of my Python script." Category: Syntax Error

Query 2: "My code runs without crashing, but the loop is skipping every other item in my list instead of all of them." Category: Logic Error

Query 3: "How can I rewrite this triple-nested loop? It takes 30 seconds to process a small dataset." Category: Optimization

Query 4: "What is the difference between an Abstract Class and an Interface in Java?" Category: Conceptual Question

Query 5: "My script is very slow, and when I tried to fix it using a library, I started getting a ModuleNotFoundError." Category: Syntax Error

### **c. Perform One-shot classification.**

#### **Prompt:**

Classify the given query into one category (Syntax Error, Logic Error, Optimization, or Conceptual Question) by first following this example — Query: "I get a ModuleNotFoundError when running my script" → Category: Syntax Error — then identify the core issue using the priority Syntax Error > Logic Error > Optimization > Conceptual Question, and output the Category with a one-sentence reason.

#### **Output:**

Query ID	Query	Category	Reasoning	
Query 1	I'm getting <code>IndentationError: expected an indented block</code> on line 12 of my Python script.	Syntax Error	This is an explicit code-breaking error that prevents the script from executing.	
Query 2	My code runs without crashing, but the loop is skipping every other item in my list instead of all of them.	Logic Error	The code executes successfully but produces an incorrect output due to an error in the program's flow.	
Query 3	How can I rewrite this triple-nested loop? It takes 30 seconds to process a small dataset.	Optimization	The query focuses on improving resource efficiency and execution time rather than fixing a crash or incorrect result.	
Query 4	What is the difference between an Abstract Class and an Interface in Java?	Conceptual Question	This is a theoretical inquiry about programming principles that does not involve a specific code failure.	
Query 5	My script is very slow, and when I tried to fix it using a library, I started getting a <code>ModuleNotFoundError</code> .	Syntax Error	Although optimization is mentioned, the <code>ModuleNotFoundError</code> is a blocking syntax-level issue that takes precedence in the priority hierarchy.	

#### d. Perform Few-shot classification.

##### Prompt:

Classify the given query into one category (Syntax Error, Logic Error, Optimization, or Conceptual Question) using these examples as guidance — `IndentationError` → Syntax Error, runs but wrong output → Logic Error, slow but correct → Optimization, theory question → Conceptual Question — apply the priority Syntax Error > Logic Error > Optimization > Conceptual Question, and output the Category with a one-sentence reason.

##### Output:

Query ID	Query	Category	Reasoning	
Query 1	I'm getting <code>IndentationError: expected an indented block</code> on line 12 of my Python script.	Syntax Error	This is an explicit code-breaking error that prevents the program from executing.	
Query 2	My code runs without crashing, but the loop is skipping every other item in my list instead of all of them.	Logic Error	The code is functional but produces incorrect results due to a flaw in the internal logic.	
Query 3	How can I rewrite this triple-nested loop? It takes 30 seconds to process a small dataset.	Optimization	The primary focus is on improving resource efficiency and execution speed rather than fixing a failure.	
Query 4	What is the difference between an Abstract Class and an Interface in Java?	Conceptual Question	This is a theoretical inquiry regarding programming principles and does not involve specific code execution issues.	
Query 5	My script is very slow, and when I tried to fix it using a library, I started getting a <code>ModuleNotFoundError</code> .	Syntax Error	Although optimization is mentioned, the <code>ModuleNotFoundError</code> is a blocking syntax-level issue that takes priority in the classification hierarchy.	

#### e. Analyse improvements in technical accuracy.

##### Output:

Zero-Shot (Base): Reliable only for explicit Syntax Errors. It struggles to distinguish Logic Errors from Optimization since both involve functional code.

One-Shot (Structure): Standardizes the output format and granularity. It introduces a single reference point to help the model align with general priority.

Few-Shot (Precision): Provides the highest accuracy by establishing clear category boundaries. It successfully teaches the priority hierarchy (Syntax > Logic > Optimization > Conceptual).

Conflict Resolution: Enables the model to resolve "Hybrid" cases by prioritizing blockers (Syntax) over secondary issues (Optimization).

#### 4. Social Media Post Categorization

##### Scenario:

A social media analytics tool must classify posts into Promotion, Complaint, Appreciation, or Inquiry.

Tasks:

##### 1. Prepare sample social media posts.

###### Prompt:

Create 5 short social media posts. Make sure there is one for each category: Promotion, Complaint, Appreciation, and Inquiry. Use emojis and hashtags like a real person would.

###### Output:

ID	Email content	Category
1	"Get 20% off all drinks today! 🍹 #DailyDeal"	Promotion
2	"My package arrived broken again... 😡 #fail"	Complaint
3	"Huge thanks to the team! You guys are the best! ✨"	Appreciation
4	"Do you guys ship to Canada? 🇨🇦 #Question"	Inquiry
5	"Another 2-hour delay? Your service is just amazing. 🤔"	Complaint

##### 2. Use Zero-shot prompting.

**Prompt:**

Classify this social media post into one category: Promotion, Complaint, Appreciation, or Inquiry.

**Output:**

ID	Email content (Post)	Category	Zero-shot Result
1	"Get 20% off all drinks today! 🍹 #DailyDeal"	Promotion	Promotion
2	"My package arrived broken again... 😡 #fail"	Complaint	Complaint
3	"Huge thanks to the team! You guys are the best! 🌟"	Appreciation	Appreciation
4	"Do you guys ship to Canada? 🇨🇦 #Question"	Inquiry	Inquiry
5	"Another 2-hour delay? Your service is just amazing. 🤔"	Complaint	Appreciation

**3. Use One-shot prompting.****Prompt:**

Classify this social media post into one category: Promotion, Complaint, Appreciation, or Inquiry.

Example: Post content: "Is this available in blue?" Answer: Inquiry

**Output:**

ID	Email content	Category	One-shot Result
1	"Get 20% off all drinks today! 🍹 #DailyDeal"	Promotion	Promotion
2	"My package arrived broken again... 😡 #fail"	Complaint	Complaint
3	"Huge thanks to the team! You guys are the best! 🌟"	Appreciation	Appreciation
4	"Do you guys ship to Canada? 🇨🇦 #Question"	Inquiry	Inquiry
5	"Another 2-hour delay? Your service is just amazing. 🤔"	Complaint	Complaint

#### 4. Use Few-shot prompting.

##### **Prompt:**

Classify this social media post into one category: Promotion, Complaint, Appreciation, or Inquiry.

Examples:

1. Post content: "Buy one get one free this weekend!" -> Answer: Promotion
2. Post content: "I am so happy with my new shoes, thank you!" -> Answer: Appreciation
3. Post content: "My order is 3 days late. Where is it?"

Task: Post content: [Insert Post Content Here]

##### **Output:**

ID	Email content (Post)	Category	Few-shot Result
1	"Get 20% off all drinks today! 🍹 #DailyDeal"	Promotion	Promotion
2	"My package arrived broken again... 😡 #fail"	Complaint	Complaint
3	"Huge thanks to the team! You guys are the best! ✨"	Appreciation	Appreciation
4	"Do you guys ship to Canada? 🇨🇦 #Question"	Inquiry	Inquiry
5	"Another 2-hour delay? Your service is just amazing. 🙄"	Complaint	Complaint

#### 4. Analyse informal language handling.

The results of Task 4 demonstrate that Few-shot Prompting is the most reliable method for understanding human emotion and tone.

- Handling Sarcasm: In social media, people often use positive words to mean negative things
- The Power of Examples: While Zero-shot logic failed by taking words literally, the Few-shot method used the provided examples to recognize that negative context (delays) and emojis signal a Complaint.
- High Consistency: By providing a "mini-training set" within the prompt, the AI maintained a 100% accuracy rate across all categories, ensuring it didn't get confused by informal slang or hashtags.