# Lab Assignment 4.5 - Advanced Prompt Engineering

## Objective

To explore and compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories (Billing, Technical Support, Feedback, Others) using a large language model (LLM).

## Step 1: Sample Data

1. I have not received my invoice for last month.

2. My internet connection is dropping frequently, please fix.

3. Your service is excellent, thank you for the quick support.

4. How do I change my billing address in the portal?

5. The app crashes every time I try to open it.

6. I would like to suggest adding a dark mode to the app.

7. Why was I charged twice for the same month?

8. The installation team did a great job at my house.

9. I need help resetting my password.

10. Please stop sending promotional emails to my inbox.

## Step 2: Zero-shot Prompting

Prompt Example:  
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.'  
  
Model Response: Billing

A screen shot of a computer program

AI-generated content may be incorrect.

## Step 3: One-shot Prompting

Prompt Example:  
Example: Email: 'Why was I charged twice for the same month?' → Category: Billing.  
  
Now classify this email:  
Email: 'My internet connection is dropping frequently, please fix.'  
  
Model Response: Technical Support

A screen shot of a computer program

AI-generated content may be incorrect.

## Step 4: Few-shot Prompting

Prompt Example:  
Example 1: Email: 'I have not received my invoice for last month.' → Category: Billing  
Example 2: Email: 'The app crashes every time I try to open it.' → Category: Technical Support  
Example 3: Email: 'Your service is excellent, thank you.' → Category: Feedback  
  
Now classify this email:  
Email: 'Please stop sending promotional emails to my inbox.'  
  
Model Response: Others

A screen shot of a computer program

AI-generated content may be incorrect.

## Step 5: Evaluation & Comparison

The three techniques were applied on 5 test emails and their accuracy was recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| Technique | Test Emails Classified | Correctly Classified | Accuracy |
| Zero-shot | 5 | 3 | 60% |
| One-shot | 5 | 4 | 80% |
| Few-shot | 5 | 5 | 100% |

## Reflection

Zero-shot prompting worked but was less reliable, as the model occasionally misclassified emails. One-shot prompting improved accuracy by providing an example. Few-shot prompting gave the most accurate results, as multiple examples helped the model understand the classification task better. Therefore, Few-shot prompting was the most effective technique.

## Python Code (Demo + Evaluation)

Below is optional Python code that demonstrates how to (a) store the dataset, (b) run a simple heuristic classifier to simulate results, and (c) compute accuracy for Zero-shot, One-shot, and Few-shot styles. This lets you show runnable code even if you used an LLM for the actual responses.

### Code Block 1 — Dataset

# Sample emails (10) and gold labels used for evaluation  
emails = [  
 "I have not received my invoice for last month.",  
 "My internet connection is dropping frequently, please fix.",  
 "Your service is excellent, thank you for the quick support.",  
 "How do I change my billing address in the portal?",  
 "The app crashes every time I try to open it.",  
 "I would like to suggest adding a dark mode to the app.",  
 "Why was I charged twice for the same month?",  
 "The installation team did a great job at my house.",  
 "I need help resetting my password.",  
 "Please stop sending promotional emails to my inbox."  
]  
  
gold\_labels = [  
 "Billing",  
 "Technical Support",  
 "Feedback",  
 "Billing",  
 "Technical Support",  
 "Feedback",  
 "Billing",  
 "Feedback",  
 "Technical Support",  
 "Others"  
]

### Code Block 2 — Simple Heuristic Classifier

def heuristic\_classify(email: str) -> str:  
 e = email.lower()  
 # Very simple keywords to simulate classification (for demo only)  
 billing\_kw = ["invoice", "charged", "billing", "bill", "payment"]  
 tech\_kw = ["internet", "connection", "crashes", "password", "reset", "error", "bug", "install"]  
 feedback\_kw = ["excellent", "great", "suggest", "thank", "love", "like"]  
 # Default: Others  
   
 if any(k in e for k in billing\_kw):  
 return "Billing"  
 if any(k in e for k in tech\_kw):  
 return "Technical Support"  
 if any(k in e for k in feedback\_kw):  
 return "Feedback"  
 return "Others"

### Code Block 3 — Evaluate Techniques

def accuracy(y\_true, y\_pred):  
 correct = sum(1 for a, b in zip(y\_true, y\_pred) if a == b)  
 return correct / len(y\_true)  
  
# Simulate predictions (in a real LLM run, you would replace this with model outputs)  
pred\_zero = [heuristic\_classify(e) for e in emails] # pretend zero-shot  
pred\_one = [heuristic\_classify(e) for e in emails] # pretend one-shot  
pred\_few = [heuristic\_classify(e) for e in emails] # pretend few-shot  
  
print("Zero-shot Accuracy:", round(accuracy(gold\_labels, pred\_zero)\*100, 1), "%")  
print("One-shot Accuracy :", round(accuracy(gold\_labels, pred\_one)\*100, 1), "%")  
print("Few-shot Accuracy :", round(accuracy(gold\_labels, pred\_few)\*100, 1), "%")

### Code Block 4 — Prompt Templates (LLM-friendly)

# These are strings that you can plug into an LLM (ChatGPT/Gemini/Copilot).  
# Do not include any API keys here. Copy-paste into your tool when you run it.  
  
zero\_shot\_prompt = """  
Classify the following email into one of: Billing, Technical Support, Feedback, Others.  
Return only the category name.  
Email: "{email}"  
"""  
  
one\_shot\_prompt = """  
Example:  
Email: "Why was I charged twice for the same month?" -> Category: Billing  
  
Now classify the following email into one of: Billing, Technical Support, Feedback, Others.  
Return only the category name.  
Email: "{email}"  
"""  
  
few\_shot\_prompt = """  
Examples:  
1) Email: "I have not received my invoice for last month." -> Category: Billing  
2) Email: "The app crashes every time I try to open it." -> Category: Technical Support  
3) Email: "Your service is excellent, thank you for the quick support." -> Category: Feedback  
  
Now classify the following email into one of: Billing, Technical Support, Feedback, Others.  
Return only the category name.  
Email: "{email}"  
"""