

Lab Assignment – 4.5

Hall Ticket No.: 2303A510E6

Name – Anushka Boora

Batch – 29

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.

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.'"

c. One-shot Prompting:

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

d. Few-shot Prompting:

- Use 3–5 labeled examples in your prompt before asking the

model to classify a new email.

e. Evaluation:

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

Screenshot of VS Code showing the code editor with the file `aac4.5.py` open. The code defines functions for zero-shot, one-shot, and few-shot classification of emails. A breakpoint is set on line 27. The status bar at the bottom shows: Line 81, Col 14, Spaces: 4, UTF-8, CRLF, Python, 3.11.5.

```
def llm_simulator(prompt):
    """
    This function simulates an LLM response.
    In real applications, this is where API calls go.
    """
    print("\n---- PROMPT SENT TO MODEL ----")
    print(prompt)
    print("-----")

    # Dummy output for assignment
    return "Predicted Category"

# 1. EMAIL CLASSIFICATION

def email_zero_shot(email):
    prompt = f"""
Classify the email into:
Billing, Technical Support, Feedback, Others
Email: {email}
Category:
"""
    return llm_simulator(prompt)

def email_one_shot(email):
    prompt = f"""
Example:
Email: My payment was deducted twice.
Category: Billing
Now classify:
Email: {email}
Category:
"""
    return llm_simulator(prompt)
```

Screenshot of VS Code showing the continuation of the code editor with the file `aac4.5.py` open. The code defines functions for zero-shot, one-shot, and few-shot classification of emails. A breakpoint is set on line 27. The status bar at the bottom shows: Line 81, Col 14, Spaces: 4, UTF-8, CRLF, Python, 3.11.5.

```
def email_one_shot(email):
    prompt = f"""
Example:
Email: My payment was deducted twice.
Category: Billing
Now classify:
Email: {email}
Category:
"""
    return llm_simulator(prompt)

def email_few_shot(email):
    prompt = f"""
Example 1:
Email: I was charged extra.
Category: Billing
Example 2:
Email: App crashes on login.
Category: Technical support
Example 3:
Email: Great service!
Category: Feedback
Now classify:
Email: {email}
Category:
"""
    return llm_simulator(prompt)

# 2. TRAVEL QUERY CLASSIFICATION

def travel_few_shot(query):
    pass
```

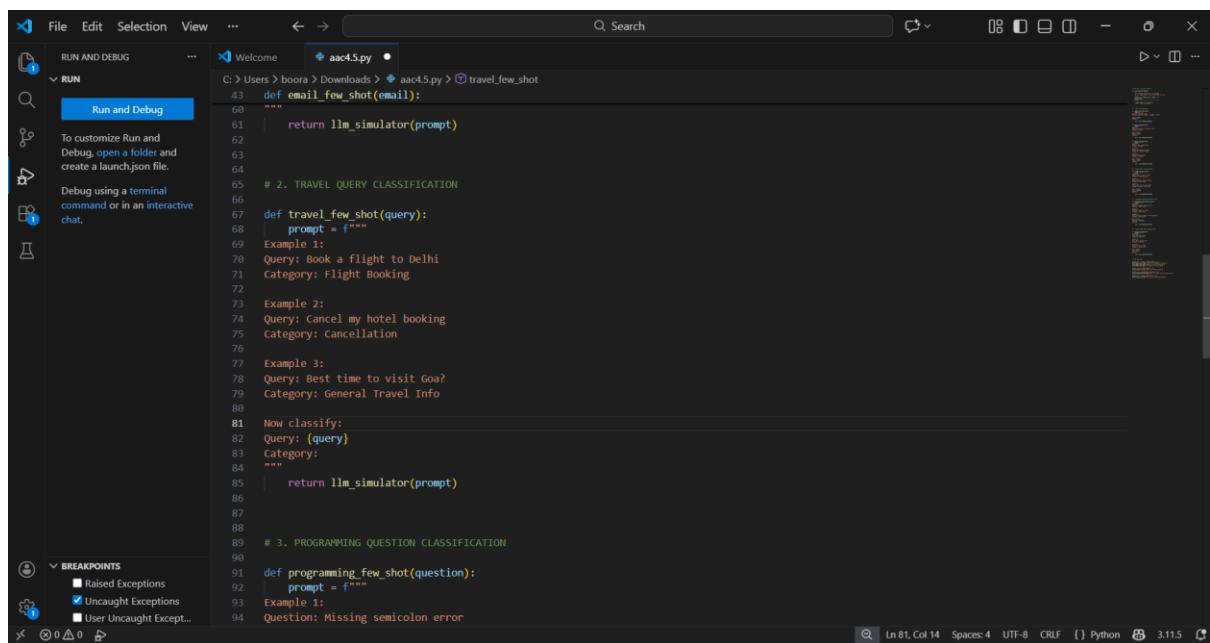
2. Travel Query Classification

Scenario:

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

Tasks:

- a. Prepare labeled travel queries.
- b. Apply Zero-shot prompting.
- c. Apply One-shot prompting.
- d. Apply Few-shot prompting.
- e. Compare response consistency.



代码编辑器显示了一个名为aac4.5.py的Python脚本，内容如下：

```
C:\> Users> boora > Downloads > aac4.5.py > travel_few_shot
43 def email_few_shot(email):
44     """
45     |     return llm_simulator(prompt)
46
47 # 2. TRAVEL QUERY CLASSIFICATION
48
49 def travel_few_shot(query):
50     """
51     prompt = f"""
52     Example 1:
53     Query: Book a flight to Delhi
54     Category: Flight Booking
55
56     Example 2:
57     Query: Cancel my hotel booking
58     Category: cancellation
59
60     Example 3:
61     Query: Best time to visit Goa?
62     Category: General Travel Info
63
64     Now classify:
65     Query: {query}
66     Category:
67     """
68
69     return llm_simulator(prompt)
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89 # 3. PROGRAMMING QUESTION CLASSIFICATION
90
91 def programming_few_shot(question):
92     """
93     prompt = f"""
94     Example 1:
95     Question: Missing semicolon error
```

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.
- b. Perform Zero-shot classification.
- c. Perform One-shot classification.
- d. Perform Few-shot classification.
- e. Analyze improvements in technical accuracy.

The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, Selection, View, and a search bar. On the left, there's a vertical toolbar with icons for Run, Debug, and other project management tools. The main area displays Python code in a code editor. A status bar at the bottom shows file information like Ln 81, Col 14, and Python version 3.11.5.

```
File Edit Selection View ... < > Search
RUN AND DEBUG ... Welcome aac4.5.py
C:\Users\boora\Downloads>aac4.5.py>travel_few_shot
67 def travel_few_shot(query):
68     Category:
69     """
70         return llm_simulator(prompt)
71
72     # 3. PROGRAMMING QUESTION CLASSIFICATION
73
74     def programming_few_shot(question):
75         prompt = f"""
76             Example 1:
77             Question: Missing semicolon error
78             Category: Syntax Error
79
80             Example 2:
81             Question: Output is wrong
82             Category: Logic Error
83
84             Example 3:
85             Question: How to reduce time complexity?
86             Category: Optimization
87
88             Now classify:
89             Question: {question}
90             Category:
91             """
92
93         return llm_simulator(prompt)
94
95     # 4. SOCIAL MEDIA POST CLASSIFICATION
96
97     def social_few_shot(post):
98         prompt = f"""
99             Example 1:
```

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.
 2. Use Zero-shot prompting.
 3. Use One-shot prompting.
 4. Use Few-shot prompting.
 5. Analyze informal language handling.

The screenshot shows the PyCharm IDE interface with the following details:

- File Menu:** File, Edit, Selection, View, ...
- Search Bar:** Search
- Toolbars:** Welcome, Run and Debug
- Run and Debug Panel:** RUN, Run and Debug. Sub-instructions: To customize Run and Debug, open a folder and create a launch.json file. Debug using a terminal command or in an interactive chat.
- Code Editor:** The code is written in Python and defines a function `social_few_shot` that takes a post as input and returns a category based on the prompt. It includes examples for promotion, complaint, appreciation, and a main execution section with travel and programming queries.
- Breakpoints:** A list of breakpoints including Raised Exceptions, Uncought Exceptions, and User Uncaught Exceptions.
- Status Bar:** Line 81, Col 14, Spaces: 4, UTF-8, CRLF, Python, 3.11.5

```
113 # 4. SOCIAL MEDIA POST CLASSIFICATION
114
115 def social_few_shot(post):
116     prompt = f"""
117     Example 1:
118     Post: Huge sale today!
119     Category: Promotion
120
121     Example 2:
122     Post: Worst service ever
123     Category: Complaint
124
125     Example 3:
126     Post: Love this brand
127     Category: Appreciation
128
129     Now classify:
130     Post: {post}
131     Category:
132     """
133     return llm_simulator(prompt)
134
135
136
137 # MAIN EXECUTION
138
139 print("\n===== EMAIL CLASSIFICATION =====")
140 print(email_zero_shot("I have not received my invoice"))
141 print(email_one_shot("Unable to reset my password"))
142 print(email_few_shot("The website is very slow"))
143
144 print("\n===== TRAVEL QUERY =====")
145 print(travel_few_shot("Cancel my flight ticket"))
146
147 print("\n===== PROGRAMMING QUESTION =====")
148
```

OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\boora\Downloads> & 'c:\Users\boora\AppData\Local\Programs\Python\Python31:5.18.0-win32-x64\bundled\libs\debugpy\launcher' '50529' '--' 'c:\Users\boora\Downloads'

===== EMAIL CLASSIFICATION =====

--- PROMPT SENT TO MODEL ---

Classify the email into:
Billing, Technical Support, Feedback, Others

Email: I have not received my invoice
Category:

-----
Predicted Category

--- PROMPT SENT TO MODEL ---

Example:
Email: My payment was deducted twice.
Category: Billing

Now classify:
Email: Unable to reset my password
Category:

-----
Predicted Category

--- PROMPT SENT TO MODEL ---

Example 1:
Email: I was charged extra.
Category: Billing

Example 2:
Email: App crashes on login.
Category: Technical Support

Example 3:
```

```
Example 1:  
Question: Missing semicolon error  
Category: Syntax Error  
  
Example 2:  
Question: Output is wrong  
Category: Logic Error  
  
Example 3:  
Question: How to reduce time complexity?  
Category: Optimization  
  
Now classify:  
Question: Why is my loop running infinitely?  
Category:  
-----  
Predicted Category  
===== SOCIAL MEDIA POST =====  
--- PROMPT SENT TO MODEL ---  
  
Example 1:  
Post: Huge sale today!  
Category: Promotion  
  
Example 2:  
Post: Worst service ever  
Category: Complaint  
  
Example 3:  
Post: Love this brand  
Example 2:  
Post: Worst service ever  
Category: Complaint  
  
Example 3:  
Post: Love this brand  
Category: Complaint  
  
Example 3:  
Post: Love this brand  
  
Example 3:  
Post: Love this brand
```