

AI Assisted Coding Ass-4.5

lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, Few-shot

E. Sai Nithin Ramanujan

2403A52L05 – B50

Scenario-1:

compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories

Prompt and Code:

```
1 #compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories
2 def classify_email(email, mode="zero"):
3     email = email.strip()
4     if mode == "zero":
5         prompt = f"""
6         Classify the email into one category:
7         Billing, Technical Support, Feedback, Others.
8         Email: "{email}"
9         Category:
10        """
11    elif mode == "one":
12        prompt = f"""
13        Email: "I was charged twice for my subscription."
14        Category: Billing
15
16        Email: "{email}"
17        Category:
18        """
19    elif mode == "few":
20        prompt = f"""
21        Email: "I was charged twice for my subscription."
22        Category: Billing
23
24        Email: "The app crashes when I open it."
25        Category: Technical Support
26
27        Email: "Great service, very satisfied!"
28        Category: Feedback
29
30        Email: "{email}"
31        Category:
32        """
33    return prompt
34 # Input email
35 email_input = "My internet disconnects frequently."
36 # Choose prompting technique: zero, one, few
37 result_prompt = classify_email(email_input, mode="few")
38 print("Generated Emails:")
39 print(result_prompt)
```

output:

```
PS D:\collage\AI-AC> python ass-4.5.py
Generated Emails:

Email: "I was charged twice for my subscription."
Category: Billing

Email: "The app crashes when I open it."
Category: Technical Support

Email: "Great service, very satisfied!"
Category: Feedback

Email: "My internet disconnects frequently."
Category:
```

Scenario-2:

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

Tasks:

- Prepare labeled travel queries.
- Apply Zero-shot prompting.
- Apply One-shot prompting.
- Apply Few-shot prompting.
- Compare response consistency.

Prompt and code:

Classify the following travel query into one category: Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

Query: "<USER_INPUT>"

```
#Task:2
def classify_travel_query(query, mode="zero"):
    query = query.strip()
    if mode == "zero":
        prompt = f"""
Classify the travel query into one category:
Flight Booking, Hotel Booking, Cancellation, General Travel Info.
Query: "{query}"
Category:
"""
    elif mode == "one":
        prompt = f"""
Query: "Book a flight from Delhi to Paris"
Category: Flight Booking

Query: "{query}"
Category:
"""
    elif mode == "few":
        prompt = f"""
Query: "Book a flight from Delhi to Paris"
Category: Flight Booking
Query: "Find hotels in Goa"
Category: Hotel Booking
Query: "Cancel my hotel reservation"
Category: Cancellation

Query: "{query}"
Category:
"""
    else:
        prompt = f"""
Classify the travel query into one category:
Flight Booking, Hotel Booking, Cancellation, General Travel Info.
Query: "{query}"
Category:
"""
    return prompt
```

```
# ♦ Terminal Input
query_input = input("Enter travel query: ")
mode_input = input("Choose prompting mode (zero / one / few): ").lower()
# Generate prompt
result_prompt = classify_travel_query(query_input, mode_input)
print("\nGenerated Prompt:")
print(result_prompt)
```

Output:

```
PS D:\collage\AI-AC> & "C:\Users\Sai Nithin\AppData\Local\Programs\Python\Python311\python.exe" d:/collage/AI-AC/ass-4.5.py
Enter travel query: What is the best time to visit Manali?
Choose prompting mode (zero / one / few): one

Generated Prompt:

Query: "Book a flight from Delhi to Paris"
Category: Flight Booking

Query: "What is the best time to visit Manali?"
Category:
```

Scenario-3: A coding help chatbot must classify queries into Syntax Error, Logic

Error, Optimization, or Conceptual Question.

Prompt and code: Classify the following programming question into one category: Syntax Error, Logic Error, Optimization, or Conceptual Question.

Question: "<USER_INPUT>"

```
#Task:3
def classify_programming_query(query, mode="zero"):
    query = query.strip()

    if mode == "zero":
        prompt = f"""
Classify the programming question into one category:
Syntax Error, Logic Error, Optimization, Conceptual Question.
Question: "{query}"
Category:
"""

    elif mode == "one":
        prompt = f"""
Question: "Why am I getting a missing semicolon error?"
Category: Syntax Error
Question: "{query}"
Category:
"""

    elif mode == "few":
        prompt = f"""
Question: "Why am I getting a missing semicolon error?"
Category: Syntax Error
Question: "My program runs but gives wrong output."
Category: Logic Error
Question: "How can I reduce time complexity?"
Category: Optimization
Question: "{query}"
Category:
"""

    return prompt
# ♦ Terminal Input
query_input = input("Enter programming question: ")
mode_input = input("Choose prompting mode (zero / one / few): ").lower()
result_prompt = classify_programming_query(query_input, mode_input)
print("\nGenerated Prompt:")
print(result_prompt)
```

Output:

```
PS D:\collage\AI-AC> & "C:\Users\Sai Nithin\AppData\Local\Programs\Python\Python311\python.exe" d:/collage/AI-AC/ass-4.5.py
Enter programming question: Why am I getting a missing semicolon error?
Choose prompting mode (zero / one / few): few

Generated Prompt:

Question: "Why am I getting a missing semicolon error?"
Category: Syntax Error

Question: "My program runs but gives wrong output."
Category: Logic Error

Question: "How can I reduce time complexity?"
Category: Optimization

Question: "Why am I getting a missing semicolon error?"
Category:
```

Scenario-4:

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

Prompt and code:

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

Post: "<USER_INPUT>"

```
#Task--4
def classify_social_post(post, mode="zero"):
    post = post.strip()
    if mode == "zero":
        prompt = f"""
Classify the social media post into one category:
Promotion, Complaint, Appreciation, Inquiry.
Post: "{post}"
Category:
"""
    elif mode == "one":
        prompt = f"""
Post: "Amazing product, loved it!"
Category: Appreciation

Post: "{post}"
Category:
"""
    elif mode == "few":
        prompt = f"""
Post: "Check out our new sale today!"
Category: Promotion

Post: "My order is delayed again."
Category: Complaint

Post: "Amazing product, loved it!"
Category: Appreciation

Post: "{post}"
Category:
"""
    return prompt
# ♦ Terminal Input
post_input = input("Enter social media post: ")
mode_input = input("Choose prompting mode (zero / one / few): ").lower()
result_prompt = classify_social_post(post_input, mode_input)
print("\nGenerated Prompt:")
print(result_prompt)
```

Output:

```
PS D:\collage\AI-AC> & "C:\Users\Sai Nithin\AppData\Local\Programs\Python\Python311\python.exe"  
d:/collage/AI-AC/ass-4.5.py  
Enter social media post: My order is delayed again  
Choose prompting mode (zero / one / few): few  
  
Generated Prompt:  
  
Post: "Check out our new sale today!"  
Category: Promotion  
  
Post: "My order is delayed again."  
Category: Complaint  
  
Post: "Amazing product, loved it!"  
Category: Appreciation  
  
Post: "My order is delayed again"  
Category:
```