

## **ASSIGNMENT-4.1**

**NAME:** L.Srinivas Reddy

**H.T.NO:**2303A52274

**BATCH:**36

### **Customer Email Classification**

A company receives a large number of customer emails every day and wants to automatically classify them into the following categories:

- Billing
- Technical Support
- Feedback
- Others

Instead of training a new machine learning model, the company decides to use prompt engineering techniques with an existing large language model.

#### Tasks

1. Prepare five short sample emails, each belonging to one of the above categories.
2. Write a zero-shot prompt to classify a given email into one of the categories without providing any examples.
3. Write a one-shot prompt by including one labeled email example and ask the model to classify a new email.
4. Write a few-shot prompt by including two or three labeled email examples and ask the model to classify a new email.
5. Compare the outputs obtained using zero-shot, one-shot, and few-shot prompting techniques and briefly comment on their effectiveness.

**CODE:**

```

def classify_email(email_text):
    email_text = email_text.lower()
    if any(keyword in email_text for keyword in ['crash', 'bug', 'error', 'not working', 'technical issue']):
        return 'Technical Support'
    elif any(keyword in email_text for keyword in ['bill', 'invoice', 'payment', 'charge']):
        return 'Billing'
    elif any(keyword in email_text for keyword in ['feedback', 'suggestion', 'improve']):
        return 'Feedback'
    else:
        return 'Others'

# Example usage:
email_to_classify = 'My app crashes every time I try to log in.'
category = classify_email(email_to_classify)
print(f"The email is classified as: {category}")

... The email is classified as: Technical Support
```

```

def classify_email(email_text):
    email_text = email_text.lower()
    if any(keyword in email_text for keyword in ['crash', 'bug', 'error', 'not working', 'technical issue']):
        return 'Technical Support'
    elif any(keyword in email_text for keyword in ['bill', 'invoice', 'payment', 'charge']):
        return 'Billing'
    elif any(keyword in email_text for keyword in ['feedback', 'suggestion', 'improve']):
        return 'Feedback'
    else:
        return 'Others'

# Example from the one-shot prompt:
email_to_classify_one_shot = 'The new update is very user-friendly.'
category_one_shot = classify_email(email_to_classify_one_shot)
print(f"The email '{email_to_classify_one_shot}' is classified as: {category_one_shot}")

... The email "The new update is very user-friendly." is classified as: Others
```

```

def classify_email(email_text):
    email_text = email_text.lower()
    if any(keyword in email_text for keyword in ['crash', 'bug', 'error', 'not working', 'technical issue']):
        return 'Technical Support'
    elif any(keyword in email_text for keyword in ['bill', 'invoice', 'payment', 'charge']):
        return 'Billing'
    elif any(keyword in email_text for keyword in ['feedback', 'suggestion', 'improve']):
        return 'Feedback'
    else:
        return 'Others'

# Example of classifying a new email:
new_email = "I'd like to suggest a new feature for your app."
classified_category = classify_email(new_email)
print(f"The email '{new_email}' is classified as: {classified_category}")

... The email "I'd like to suggest a new feature for your app." is classified as: Others
```

```

def classify_email(email_text):
    email_text = email_text.lower()
    if any(keyword in email_text for keyword in ['crash', 'bug', 'error', 'not working', 'technical issue']):
        return 'Technical Support'
    elif any(keyword in email_text for keyword in ['bill', 'invoice', 'payment', 'charge']):
        return 'Billing'
    elif any(keyword in email_text for keyword in ['feedback', 'suggestion', 'improve']):
        return 'Feedback'
    else:
        return 'Others'

# Example from the few-shot prompt:
email_to_classify_few_shot = 'Can you tell me your office working hours?'
category_few_shot = classify_email(email_to_classify_few_shot)
print(f"The email '{email_to_classify_few_shot}' is classified as: {category_few_shot}")

... The email "Can you tell me your office working hours?" is classified as: Others
```

## Intent Classification for Chatbot Queries

A company wants to deploy a chatbot to handle customer queries.

Each query must be classified into one of the following intents:

Account Issue, Order Status, Product Inquiry, or General Question  
using prompt engineering techniques.

Tasks to be Completed

### 1. Prepare Sample Data

Create 6 short chatbot user queries, each mapped to one of the four intents.

## 2. Zero-shot Prompting

Design a prompt that asks the LLM to classify a user query into the given intent categories without examples.

## 3. One-shot Prompting

Provide one labeled query in the prompt before classifying a new query.

## 4. Few-shot Prompting

Include 3–5 labeled intent examples to guide the LLM before classifying a new query.

## 5. Evaluation

Apply all three techniques to the same set of test queries and document differences in performance.

### CODE:

```
⌚ def classify_query(query_text):
    query_text = query_text.lower()
    if any(keyword in query_text for keyword in ['account', 'login', 'log in', 'password', 'locked']):
        return 'Account Issue'
    elif any(keyword in query_text for keyword in ['order', 'package', 'delivery', 'arrive']):
        return 'Order Status'
    elif any(keyword in query_text for keyword in ['product', 'phone', 'laptop', 'gaming', 'features', 'specifications']):
        return 'Product Inquiry'
    else:
        return 'General Question'

# Example from the zero-shot prompt:
query_to_classify = 'Where is my order?'
intent = classify_query(query_to_classify)
print(f"The query '{query_to_classify}' is classified as: {intent}")

... The query "Where is my order?" is classified as: Order Status
```

```
⌚ def classify_query(query_text):
    query_text = query_text.lower()
    if any(keyword in query_text for keyword in ['account', 'login', 'log in', 'password', 'locked']):
        return 'Account Issue'
    elif any(keyword in query_text for keyword in ['order', 'package', 'delivery', 'arrive']):
        return 'Order Status'
    elif any(keyword in query_text for keyword in ['product', 'phone', 'laptop', 'gaming', 'features', 'specifications']):
        return 'Product Inquiry'
    else:
        return 'General Question'

# Example from the one-shot prompt:
query_to_classify_one_shot = 'Does this phone support 5G?'
intent_one_shot = classify_query(query_to_classify_one_shot)
print(f"The query '{query_to_classify_one_shot}' is classified as: {intent_one_shot}")

... The query "Does this phone support 5G?" is classified as: Product Inquiry
```

```
⌚ def classify_query(query_text):
    query_text = query_text.lower()
    if any(keyword in query_text for keyword in ['account', 'login', 'log in', 'password', 'locked']):
        return 'Account Issue'
    elif any(keyword in query_text for keyword in ['order', 'package', 'delivery', 'arrive']):
        return 'Order Status'
    elif any(keyword in query_text for keyword in ['product', 'phone', 'laptop', 'gaming', 'features', 'specifications']):
        return 'Product Inquiry'
    else:
        return 'General Question'

# Example from the few-shot prompt:
query_to_classify_few_shot = 'What are your business hours?'
intent_few_shot = classify_query(query_to_classify_few_shot)
print(f"The query '{query_to_classify_few_shot}' is classified as: {intent_few_shot}")

... The query "What are your business hours?" is classified as: General Question
```

```
▶ def classify_query(query_text):
    query_text = query_text.lower()
    if any(keyword in query_text for keyword in ['account', 'login', 'log in', 'password', 'locked']):
        return 'Account Issue'
    elif any(keyword in query_text for keyword in ['order', 'package', 'delivery', 'arrive']):
        return 'Order Status'
    elif any(keyword in query_text for keyword in ['product', 'phone', 'laptop', 'gaming', 'features', 'specifications']):
        return 'Product Inquiry'
    else:
        return 'General Question'

# Example of classifying a new query:
new_query = "I need help resetting my password."
classified_intent = classify_query(new_query)
print(f"The query '{new_query}' is classified as: {classified_intent}")
... The query "I need help resetting my password." is classified as: Account Issue
```