

AI Assisted Coding

Lab Assignment 4.5

Name : T. Goutham Reddy

Hall Ticket no : 2303A1345

Batch No : 20

Task -1:

Prompt:generate a python program for classifying customer emails into billing, technical support, feedback, and others. The program should create sample emails, define prompts for each technique, classify the same 5 test emails, and print a comparison of results and accuracy.

```
from typing import List, Tuple, Dict

# Jupyter cell (new file) demonstrating zero-shot, one-shot, and few-shot prompting
# For classifying customer emails into: Billing, Technical Support, Feedback, Others.
# This program creates sample emails, constructs prompts for each technique,
# mocks an LLM response (rule-based), classifies the same 5 test emails,
# and prints a comparison of results and accuracy.

# Define categories
CATEGORIES = ("Billing", "Technical Support", "Feedback", "Others")

# Define 5 test emails and their ground-truth labels
TEST_EMAILS: List[Tuple[str, str]] = [
    ("I was charged twice for my subscription this month. Please refund the extra charge.", "Billing"),
    ("My app crashes whenever I try to upload a photo. It shows an unexpected error.", "Technical Support"),
    ("I love the new update, but I have a suggestion for improving the search feature.", "Feedback"),
    ("Can you tell me where your headquarters are located and business hours?", "Others"),
    ("I cannot log in after the password reset, it says invalid token.", "Technical Support"),
]

# Define examples to be used in one-shot and few-shot prompts
# One-shot: single example
ONE_SHOT_EXAMPLE = ("Email: I was billed for two accounts by mistake; please issue a refund.\n\nLabel: Billing")

# Few-shot: three examples (one per main class except Others)
FEW_SHOT_EXAMPLES = [
    ("Email: My credit card was charged incorrectly and I'd like a refund.\n\nLabel: Billing"),
    ("Email: The app shows an error and won't let me complete checkout.\n\nLabel: Technical Support"),
    ("Email: Great job on the redesign! One suggestion: add filters to search.\n\nLabel: Feedback"),
]

# Instruction template for the classifier
INSTRUCTION = f"""
    Classify the following customer email into one of these categories:
    {CATEGORIES}.
    Respond with a single label.
"""

# Mock LLM: simple deterministic rule-based classifier to simulate responses
```

Output :

Index	Email (short)	Truth	Zero-shot	One-shot	Few-shot
1	I was charged twice for my subscription ...	Billing	Billing	Billing	Billing
2	My app crashes whenever I try to upload ...	Technical Support	Billing	Billing	Billing
3	I love the new update, but I have a sugg...	Feedback	Billing	Billing	Billing
4	Can you tell me where your headquarters are located and business hours?	Others	Billing	Billing	Billing
5	I cannot log in after the password reset...	Technical Support	Billing	Billing	Billing

Accuracies:

Zero-shot accuracy: 20.00%

One-shot accuracy: 20.00%

Few-shot accuracy: 20.00%

Task 2:

Prompt: generate a python program that classify travel queries into flight booking, hotel booking, cancellation, and general travel information. The program should create labeled travel queries, apply all three prompting techniques .

```
class Solution {
public:
    vector<string> findLadders(string beginWord, string endWord, vector<string> wordList) {
        if (!beginWord || !endWord || !wordList.size()) return {};
        unordered_set<string> words(wordList.begin(), wordList.end());
        if (words.find(beginWord) == words.end()) words.insert(beginWord);
        if (words.find(endWord) == words.end()) words.insert(endWord);
        queue<string> q{{beginWord}};
        int level = 0;
        while (!q.empty() && words[beginWord] != words[endWord]) {
            level++;
            int size = q.size();
            for (int i = 0; i < size; ++i) {
                string curr = q.front();
                q.pop();
                for (int j = 0; j < curr.length(); ++j) {
                    char originalChar = curr[j];
                    for (char c = 'a'; c <= 'z'; ++c) {
                        curr[j] = c;
                        if (curr == endWord) return {beginWord, curr};
                        if (words[curr]) {
                            words.erase(curr);
                            q.push(curr);
                        }
                    }
                    curr[j] = originalChar;
                }
            }
        }
        return {};
    }
};
```

Output:

Idx	Query (short)	Truth	Zero-shot	One-shot	Few-shot
1	I need to book a round-trip ticket from ...	Flight Booking	Cancellation	Cancellation	Cancellation
2	Can I cancel my hotel reservation and go...	Cancellation	Cancellation	Cancellation	Cancellation
3	What time is check-in at your spontaneou...	Hotel Booking	Cancellation	Cancellation	Cancellation
4	Do I need a visa to travel to Japan as a ...	General Travel Info	Cancellation	Cancellation	Cancellation
5	My flight was delayed and I want to chan...	Flight Booking	Cancellation	Cancellation	Cancellation
6	How much is baggage allowance for intern...	General Travel Info	Cancellation	Cancellation	Cancellation

Accuracies:

Zero-shot accuracy: 36.87%

One-shot accuracy: 36.87%

Few-shot accuracy: 36.87%

Consistency among prompting techniques:

Zero-shot vs One-shot agreement: 100.00%

Zero-shot vs Few-shot agreement: 100.00%

One-shot vs Few-shot agreement: 100.00%

All three equal rates: 100.00%

Task 3:

Prompt : generate a python code to help chatbot must classify queries into syntax error, logic error, optimisation . The program should also include coding-related user queries,Perform zero-shot , one-shot , few-shot classification and Analyse improvements in technical accuracy.

```
generate a python code to help chatbot must classify queries into syntax error, logic error, optimisation . The program should also include coding-related user queries,Perform zero-shot , one-shot , few-shot classification and give me the improvements in technical accuracy.
```

Ask or edit in context

Accept Close Accept & Run (0) []

```
# This will take multiple user inputs and return the appropriate response
# TEST_VARIABLE_CODE : INSTRUCTION_CODE, ONE_SHOT_EXAMPLE_CODE, FEW_SHOT_EXAMPLES_CODE

def build_prompt_zero_shot_code(query: str) -> str:
    return f'{INSTRUCTION_CODE}{{query}} {query}{VALLEY}'

def build_prompt_one_shot_code(query: str, example_code: str) -> str:
    return f'{example_code}{VALLEY}{INSTRUCTION_CODE}{query}{VALLEY}'

def build_prompt_for_shot_code(query: str, examples: dict) -> str:
    examples_text = "\n".join(examples)
    return f'{examples_text}{VALLEY}{INSTRUCTION_CODE}{{query}}{VALLEY}'

def mock_llm_code_response(prompt: str) -> str:
    # Implement a simple keyword-based classifier for coding queries
    lower = prompt.lower()

    syntax_kw = [
        "syntaxerror", "syntax error", "invalid syntax", "unexpected left", "unexpected indent",
        "missing colon", "parse error", "indentationerror", "syntaxerror"
    ]

    logic_kw = [
        "run-time error", "runtime error", "never ends", "wrong output", "missing code", "bug",
        "timeout", "index error", "out of range", "wrong result", "incorrect"
    ]

    optimization_kw = [
        "slow", "too slow", "optimal", "optimise", "performance", "optimality", "memory",
        "time", "speed", "big inputs", "reduce memory", "faster", "optimise"
    ]

    conceptual_kw = [
        "reference informed", "what is the", "what is", "definition", "how does", "concept"
    ]
```

Output :

Idx	Query (short)	Truth	Zero-shot	One-shot	Few-shot
3	Why do I get SyntaxError: Invalid syntax...	Syntax Error	Syntax Error	Syntax Error	Syntax Error
2	My loop never ends because I increment t...	Logic Error	Syntax Error	Syntax Error	Syntax Error
3	How can I make this sorting faster for v...	Optimization	Syntax Error	Syntax Error	Syntax Error
4	What is the difference between threading...	Conceptual Question	Syntax Error	Syntax Error	Syntax Error
5	My function returns wrong sum for some ...	Logic Error	Syntax Error	Syntax Error	Syntax Error
6	I see IndexError: list index out of range	Logic Error	Syntax Error	Syntax Error	Syntax Error
7	Why does my recursion hit maximum recurs...	Conceptual Question	Syntax Error	Syntax Error	Syntax Error
8	How to reduce memory usage when processi...	Optimization	Syntax Error	Syntax Error	Syntax Error

Accuracies:

```

Zero-shot accuracy: 12.50%
One-shot accuracy: 12.50% (Improvement over zero-shot: 0.00%)
Few-shot accuracy: 12.50% (Improvement over zero-shot: 0.00%)
Few-shot vs One-shot Improvement: 0.00%

```

Improvements (Indices, #-based):

```

Zero-shot => One-shot: Improved on examples: []
Zero-shot => Few-shot: Improved on examples: []
One-shot => Few-shot: Improved on examples: []

```

Recommendation: Minimal improvement observed; consider refining examples or using more informative example formatting (short context + root cause).

Task 4 :

Prompt : generate a python program to classify social media posts into Promotion, Complaint, Appreciation, or Inquiry using zero-shot, one-shot, and few-shot prompting, and compare informal language handling.

```

# Social media post classification using zero-shot, one-shot, few-shot prompting; and compare informal language handling.

● Write a Python program to classify social media posts into Promotion, Complaint, Appreciation, or Inquiry using zero-shot, one-shot, and few-shot prompting; and compare informal language handling.

Ask or edit in context
Accept Close Accept & Run ⌘ + ⌘ OPT-5 sec - ➔
# This 20-line script classifies and labels informal text posts using zero-shot LLMs
# Compares informal accuracy and informal language handling.

SOCIAL_CATEGORIES = ("Promotion", "Complaint", "Appreciation", "Inquiry")

INSTRUCTION_SOCIAL = """
Classify the following social media post. Make one of these categories: 
"Promotion", "Complaint", "Appreciation", "Inquiry". Respond with a single label.
"""

# Test posts (some contain informal language, some not)
TEST_POSTS = [
    ("Huge SALE this weekend!! 50% off all items, buy now!!", "Promotion"),
    ("My new phone arrived dead. Not happy, wait refund.", "Complaint"),
    ("Thanks team, app is working great! 🎉", "Appreciation"),
    ("How do I change my password? Pls help.", "Inquiry"),
    ("Wow the customer service was amazing, thx! 🙏", "Appreciation"),
    ("Any promo codes for students? plz?", "Inquiry"),
    ("Coupon code not applied, still charged full price. Disappointed!", "Complaint"),
    ("Free giveaway! RT to enter #contest", "Promotion"),
    ("Why why the app crashes when I try to log in?", "Complaint"),
    ("Are you open on Sundays?", "Inquiry"),
]

# One-shot example (some posts are labeled correctly)
ONE_SHOT_EXAMPLE_SOCIAL = "Post: Get 20% off with code SWI20; amazing deals! Label: Promotion"

FEW_SHOT_EXAMPLES_SOCIAL = [
    "Post: I was killed twice for my purchase, please refund. Label: Complaint",
    "Post: Love the new update, great work team! 🎉 Label: Appreciation"
]

```

Output :

```

> 0.06
Python

idx | Post (short) | Truth | zero-shot | one-shot | few-shot
---|---|---|---|---|---
1 | Huge SALE this weekend!! 50% off all items... | Promotion | Promotion | Promotion | Promotion
2 | My new phone arrived dead. Not happy, wa... | Complaint | Promotion | Promotion | Promotion
3 | Thanks team, app is working great! 🎉 | Appreciation | Promotion | Promotion | Promotion
4 | How do I change my password? Pls help. | Inquiry | Promotion | Promotion | Promotion
5 | Wow the customer service was amazing, thx! 🙏 | Appreciation | Promotion | Promotion | Promotion
6 | Any promo codes for students? plz? | Inquiry | Promotion | Promotion | Promotion
7 | Coupon code not applied, still charged f... | Complaint | Promotion | Promotion | Promotion
8 | Free giveaway! RT to enter #contest | Promotion | Promotion | Promotion | Promotion
9 | Why why the app crashes when I try to lo... | Complaint | Complaint | Complain | Complaint
10 | Are you open on Sundays? | Inquiry | Promotion | Promotion | Promotion

Accuracies:
zero-shot: 38.00%  one-shot: 38.00%  few-shot: 38.00%

Informal-language subset accuracies:
Zero-shot (informal): 37.50%  One-shot (informal): 37.50%  Few-shot (informal): 37.50%

Consistency:
ZS vs OS agreement: 100.00%
ZS vs PS agreement: 100.00%
OS vs PS agreement: 100.00%

Recommendation: Few-shot examples help the model better handle informal phrasing. Add diverse informal examples.

```