SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s)Name		1. Dr. Mohammed Ali Shaik 2. Dr. T Sampath Kumar 3. Mr. S Naresh Kumar 4. Dr. V. Rajesh 5. Dr. Brij Kishore 6. Dr Pramoda Patro 7. Dr. Venkataramana 8. Dr. Ravi Chander 9. Dr. Jagjeeth Singh	
Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	06-08-2025	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	
AssignmentNumber:4.5(Present assignment number)/**24**(Total number of assignments)			

Q. No.	Question	*ExpectedTime to complete*
1	**Lab 4: Advanced Prompt Engineering: Zero-shot, one-shot, and few-shot techniques** **Objective:** To explore and compare Zero-shot, One-shot, and Few-shot prompting techniques for classifying emails into predefined categories using a large language model (LLM). 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 1. **Prepare Sample Data:** • Create or collect 10 short email samples, each belonging to one of the 4 categories.	08.08.2025 EOD

```
email_samples = [
    {"category": "Work", "email": "Subject: Meeting Reminder\nHi Team, Just a reminder about our meeting tomorrow at 10 AM in Room 3B. Please be prepared to discuss the Q3 projections. Thanks, [Your Name]"},
    {"category": "Work", "email": "Subject: Project Update Request\nHello [Colleague's Name], Could you please provide an update on the status of the Alpha Project by end of day today? We need to finalize the report. Best, [Your Na
    {"category": "Work", "email": "Subject: Vacation Request\nDear [Manager's Name], I would like to request vacation leave from [Start Date] to [End Date]. Please let me know if this works. Thank you, [Your Name]"},
    {"category": "Spam", "email": "Subject: Urgent: Claim Your Prize Now!\nCongratulations! You've been selected to receive a free gift card. Click here to claim your prize now: [Malicious Link]"},
    {"category": "Spam", "email": "Subject: Important Account Notification\nDear User, Your account has been compromised. Please verify your details by clicking on the link below: [Malicious Link]"},
    {"category": "Spam", "email": "Subject: Exclusive Offer - Limited Time!\nGet 50% off on all products! Shop now before the offer expires: [Malicious Link]"},
    {"category": "Promotions", "email": "Subject: New Arrivals This Week!\nCheck out our latest collection of [Product Type]. Shop now and get free shipping on orders over $50! [Link to Website]"},
    {"category": "Promotions", "email": "Subject: Don't Miss Out on Our Sale!\nOur annual summer sale is here! Enjoy amazing discounts on your favorite items. Visit our website today! [Link to Website]"},
    {"category": "Promotions", "email": "Subject: Earn Double Points This Weekend!\nLoyalty members, earn double points on all purchases this weekend. Not a member yet? Join now! [Link to Website]"},
    {"category": "Social", "email": "Subject: Photos from the Weekend!\nHey [Friend's Name], Check out the photos from our weekend trip! They turned out great. [Link to Photo Album]"}
]

import pandas as pd

df = pd.DataFrame(email_samples)
display(df)
```

1 to 10 of 10 entries Filter

index	category	email
2	Work	Subject: Vacation Request Dear [Manager's Name], I would like to request vacation leave from [Start Date] to [End Date]. Please let me know if this works. Thank you. [Your Name]
3	Spam	Subject: Urgent: Claim Your Prize Now! Congratulations! You've been selected to receive a free gift card. Click here to claim your prize now: [Malicious Link]
1	Work	Subject: Project Update Request Hello [Colleague's Name], Could you please provide an update on the status of the Alpha Project by end of day today? We need to finalize the report. Best, [Your Name]
9	Social	Subject: Photos from the Weekend! Hey [Friend's Name], Check out the photos from our weekend trip! They turned out great. [Link to Photo Album]
6	Promotions	Subject: New Arrivals This Week! Check out our latest collection of [Product Type]. Shop now and get free shipping on orders over $50! [Link to Website]
0	Work	Subject: Meeting Reminder Hi Team, Just a reminder about our meeting tomorrow at 10 AM in Room 3B. Please be prepared to discuss the Q3 projections. Thanks, [Your Name]
4	Spam	Subject: Important Account Notification Dear User, Your account has been compromised. Please verify your details by clicking on the link below: [Malicious Link]
5	Spam	Subject: Exclusive Offer - Limited Time! Get 50% off on all products! Shop now before the offer expires: [Malicious Link]
8	Promotions	Subject: Earn Double Points This Weekend! Loyalty members, earn double points on all purchases this weekend. Not a member yet? Join now! [Link to Website]
7	Promotions	Subject: Don't Miss Out on Our Sale! Our annual summer sale is here! Enjoy amazing discounts on your favorite items. Visit our website today! [Link to Website]

Show 25 per page

2. **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.'"

+ Code + Text

```
email_to_classify = df['email'][0] # Using the first email from the dataframe as an example

zero_shot_prompt = f"""Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''{email_to_classify}'''

Category:
"""

print(zero_shot_prompt)
```

```
Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''Subject: Meeting Reminder
Hi Team, Just a reminder about our meeting tomorrow at 10 AM in Room 3B. Please be prepared to discuss the Q3 projections. Thanks, [Your Name]'''

Category:
```

3. One-shot Prompting:

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

```
example_email = df['email'][9]  # Using the last email from the dataframe as an example
example_category = df['category'][9]

email_to_classify_oneshot = df['email'][0] # Using the first email from the dataframe as the one to classify

one_shot_prompt = f"""Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''{example_email}'''
Category: {example_category}

Email: '''{email_to_classify_oneshot}'''
Category:
"""

print(one_shot_prompt)
```

```
Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''Subject: Photos from the Weekend!
Hey [Friend's Name], Check out the photos from our weekend trip! They turned out great. [Link to Photo Album]'''
Category: Social

Email: '''Subject: Meeting Reminder
Hi Team, Just a reminder about our meeting tomorrow at 10 AM in Room 3B. Please be prepared to discuss the Q3 projections. Thanks, [Your Name]'''
Category:
```

4. **Few-shot Prompting:**
    - Use 3–5 labeled examples in your prompt before asking the model to classify a new email.

```python
# Using 3 examples from the dataframe for few-shot prompting
example_email_1 = df['email'][0]
example_category_1 = df['category'][0]

example_email_2 = df['email'][3]
example_category_2 = df['category'][3]

example_email_3 = df['email'][6]
example_category_3 = df['category'][6]

email_to_classify_fewshot = df['email'][9] # Using a different email to classify

few_shot_prompt = f"""Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''{example_email_1}'''
Category: {example_category_1}

Email: '''{example_email_2}'''
Category: {example_category_2}

Email: '''{example_email_3}'''
Category: {example_category_3}

Email: '''{email_to_classify_fewshot}'''
Category:
"""

print(few_shot_prompt)
```

```
Classify the following email into one of the following categories: Work, Spam, Promotions, Social.

Email: '''Subject: Meeting Reminder
Hi Team, Just a reminder about our meeting tomorrow at 10 AM in Room 3B. Please be prepared to discuss the Q3 projections. Thanks, [Your Name]'''
Category: Work

Email: '''Subject: Urgent: Claim Your Prize Now!
Congratulations! You've been selected to receive a free gift card. Click here to claim your prize now: [Malicious Link]'''
Category: Spam

Email: '''Subject: New Arrivals This Week!
Check out our latest collection of [Product Type]. Shop now and get free shipping on orders over $50! [Link to Website]'''
Category: Promotions

Email: '''Subject: Photos from the Weekend!
Hey [Friend's Name], Check out the photos from our weekend trip! They turned out great. [Link to Photo Album]'''
Category:
```

5. **Evaluation:**

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

```python
import pandas as pd

# 1. Define Test Emails
test_emails = [
    {"category": "Billing", "email": "Subject: Your monthly bill is ready\nDear Customer, Your latest bill is now available. You can view it online here: [Link to Billing Portal]"},
    {"category": "Technical Support", "email": "Subject: Issue with software installation\n\nHi Support Team, I'm having trouble installing the software update. I keep getting an error message. Can you help? Thanks, [Your Name]"},
    {"category": "Feedback", "email": "Subject: Feedback on your recent service\n\nDear [Company Name], I wanted to provide some feedback on my recent interaction with your service. Overall, I was satisfied. [Further Details]"},
    {"category": "Billing", "email": "Subject: Payment Reminder\n\nJust a friendly reminder that your payment is due soon. Please make sure to pay by [Due Date] to avoid any interruptions. Thank you."},
    {"category": "Others", "email": "Subject: Newsletter Subscription Confirmation\n\nThank you for subscribing to our newsletter! You'll receive updates on our latest news and offers. [Link to manage subscription]"}
]

df_test = pd.DataFrame(test_emails)

# 2. Implement Prompting Techniques (using a simulated model) and 3. Collect Results
# This is a simulated model prediction function. Replace with your actual LLM call.
def simulate_model_prediction(prompt, few_shot_examples=None):
    """
    Simulates a large language model's prediction based on the prompt.
    For few-shot, it returns the category of the last example.
    For zero-shot and one-shot, it tries to find a category from the prompt or email content.
    """
    categories = ["Billing", "Technical Support", "Feedback", "Others"]

    if few_shot_examples:
        # For few-shot, return the category of the last example from the few_shot_examples list
        return few_shot_examples[-1]["category"]
    else:
        # For zero-shot and one-shot, try to find a category in the email content
        for category in categories:
            if category.lower() in prompt.lower():
                return category
        # If no category found, return a default or None
        return "Others" # Defaulting to 'Others' if no specific category is found
```

```python
results = []

# Prepare few-shot examples from the training data (first 3 for simplicity)
# Assuming 'df' from previous cells contains the training data
if 'df' in locals():
    few_shot_examples = df.sample(3).to_dict('records')
else:
    few_shot_examples = [
        {"category": "Work", "email": "Subject: Meeting\nDetails inside."},
        {"category": "Spam", "email": "Subject: Win a prize\nClick here."},
        {"category": "Promotions", "email": "Subject: Sale!\nShop now."}
    ]


for index, row in df_test.iterrows():
    email_to_classify = row['email']
    true_category = row['category']

    # Zero-shot prompt
    zero_shot_prompt = f"""Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others.

Email: '''{email_to_classify}'''

Category:
"""
    zero_shot_prediction = simulate_model_prediction(zero_shot_prompt)

    # One-shot prompt (using a random example from the training data)
    if 'df' in locals():
      one_shot_example = df.sample(1).to_dict('records')[0]
    else:
      one_shot_example = {"category": "Social", "email": "Subject: Party pics\nCheck them out."}


    one_shot_prompt = f"""Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others.

Email: '''{one_shot_example['email']}'''
Category: {one_shot_example['category']}
```

```
Email: '''{email_to_classify}'''
Category:
"""
    one_shot_prediction = simulate_model_prediction(one_shot_prompt)

    # Few-shot prompt (using the prepared few-shot examples)
    few_shot_prompt_text = f"""Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others.
"""
    for example in few_shot_examples:
        few_shot_prompt_text += f"""
Email: '''{example['email']}'''
Category: {example['category']}
"""
    few_shot_prompt_text += f"""
Email: '''{email_to_classify}'''
Category:
"""
    few_shot_prediction = simulate_model_prediction(few_shot_prompt_text, few_shot_examples)


    results.append({
        'email': email_to_classify,
        'true_category': true_category,
        'zero_shot_prediction': zero_shot_prediction,
        'one_shot_prediction': one_shot_prediction,
        'few_shot_prediction': few_shot_prediction
    })

results_df = pd.DataFrame(results)

# 4. Compare and Document (Displaying results in a DataFrame)
display(results_df)

# Further analysis and documentation can be done in subsequent cells or markdown.
```

1 to 5 of 5 entries | Filter

index	email	true_category	zero_shot_prediction	one_shot_prediction	few_shot_prediction
0	Subject: Your monthly bill is ready Dear Customer, Your latest bill is now available. You can view it online here: [Link to Billing Portal]	Billing	Billing	Billing	Work
1	Subject: Issue with software installation Hi Support Team, I'm having trouble installing the software update. I keep getting an error message. Can you help? Thanks, [Your Name]	Technical Support	Billing	Billing	Work
2	Subject: Feedback on your recent service Dear [Company Name], I wanted to provide some feedback on my recent interaction with your service. Overall, I was satisfied. [Further Details]	Feedback	Billing	Billing	Work
3	Subject: Payment Reminder Just a friendly reminder that your payment is due soon. Please make sure to pay by [Due Date] to avoid any interruptions. Thank you.	Billing	Billing	Billing	Work
4	Subject: Newsletter Subscription Confirmation Thank you for subscribing to our newsletter! You'll receive updates on our latest news and offers. [Link to manage subscription]	Others	Billing	Billing	Work

Show 25 per page

1

I like what you see? Visit the data table notebook to learn more about interactive tables.

Requirements:

- VS Code with Github Copilot or Cursor IDE and/or Google Colab with Gemini

Deliverables:

- A .txt or .md file showing prompts and model responses.
- A comparison table showing classification accuracy for each technique.
- A short reflection on which method was most effective and why

.