

Lab Assignment-4.1

CH.vishnu vardhan

2303A51718

Batch-11

Problem Statement: A news aggregation platform wants to automatically categorize headlines into Politics, Sports, Technology, and Entertainment without training a machine learning model. Tasks to be Completed

1. Prepare Sample Data

Collect 10 news headlines, each belonging to one of the four categories.

PROMPT: #write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only this four categories dont use other categories and store them in a dictionary where the key is category and value is headline

Code:

```
palindrome.py > ...
1  #WRITE A CODE TO COLLECT 10 NEWS HEADLINES EACH BELONGING TO DIFFERENT CATEGORIES SUCH AS TECHNOLOGY,SPORTS,HEALTH,ENTERTAINMENT, POLITICS
2  news_headlines = {
3      "technology": "New AI technology is transforming the tech industry",
4      "sports": "Local team wins championship after a thrilling final",
5      "health": "New breakthrough in cancer research offers hope",
6      "entertainment": "Blockbuster movie breaks box office records",
7      "politics": "Elections bring significant changes to the government",
8      "technology_2": "Tech giant releases latest smartphone model",
9      "sports_2": "Star athlete sets new world record",
10     "health_2": "Study reveals benefits of a balanced diet",
11     "entertainment_2": "Famous singer announces world tour",
12     "politics_2": "New policies aim to improve education system"
13 }
14 # Example usage
15 for category, headline in news_headlines.items():
16     print(f"{category.capitalize()}: {headline}")
17 |
```

Output:

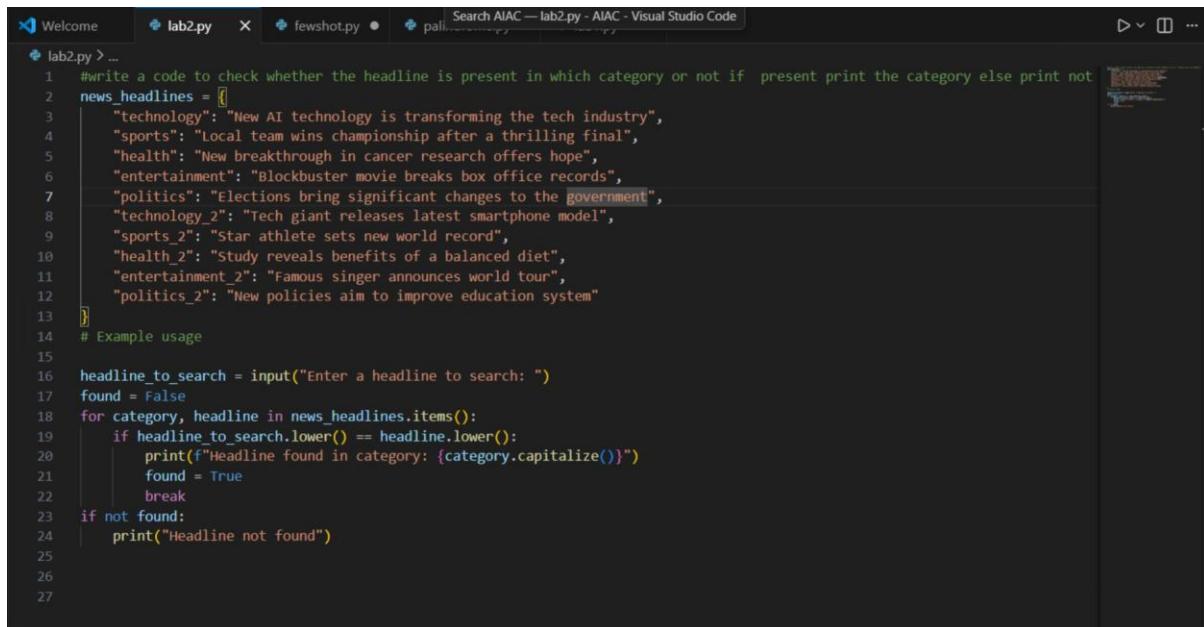
```
Sports: Local team wins championship after a thrilling final
Health: New breakthrough in cancer research offers hope
Entertainment: Blockbuster movie breaks box office records
Politics: Elections bring significant changes to the government
Technology_2: Tech giant releases latest smartphone model
Sports_2: Star athlete sets new world record
Health_2: Study reveals benefits of a balanced diet
Entertainment_2: Famous singer announces world tour
Politics_2: New policies aim to improve education system
```

2. Zero-shot Prompting

Write a prompt asking the LLM to classify a headline into a category without examples.

Prompt: #write a code to check whether the headline is present in which category or not if present print the category else print not found by using input user

Code:



```
# Welcome  lab2.py  fewshot.py  pall...  Search AIAC — lab2.py - AIAC - Visual Studio Code
lab2.py > ...
1  #write a code to check whether the headline is present in which category or not if present print the category else print not
2  news_headlines = [
3      "technology": "New AI technology is transforming the tech industry",
4      "sports": "Local team wins championship after a thrilling final",
5      "health": "New breakthrough in cancer research offers hope",
6      "entertainment": "Blockbuster movie breaks box office records",
7      "politics": "Elections bring significant changes to the government",
8      "Technology_2": "Tech giant releases latest smartphone model",
9      "Sports_2": "Star athlete sets new world record",
10     "Health_2": "Study reveals benefits of a balanced diet",
11     "Entertainment_2": "Famous singer announces world tour",
12     "Politics_2": "New policies aim to improve education system"
13 ]
14 # Example usage
15
16 headline_to_search = input("Enter a headline to search: ")
17 found = False
18 for category, headline in news_headlines.items():
19     if headline_to_search.lower() == headline.lower():
20         print(f"Headline found in category: {category.capitalize()}")
21         found = True
22         break
23 if not found:
24     print("Headline not found")
25
26
27
```

Output:

```
Enter a headline to search: Elections bring significant changes to the government
Headline found in category: Politics
```

3. One-shot Prompting

Add one labeled headline example before classifying a new headline.

PROMPT: #write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only this four categories dont use other categories and store them in a dictionary where the key is category and value

#input:"Elections bring significant changes to the government",output:"Headline found in category:Politics"

CODE:

```
palindrome.py > ...
1  #write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics
2  news_headlines = {
3      "technology": "New AI technology is transforming the tech industry",
4      "sports": "Local team wins championship after a thrilling final",
5      "health": "New breakthrough in cancer research offers hope",
6      "entertainment": "Blockbuster movie breaks box office records",
7      "politics": "Elections bring significant changes to the government",
8      "technology_2": "Tech giant releases latest smartphone model",
9      "sports_2": "Star athlete sets new world record",
10     "health_2": "Study reveals benefits of a balanced diet",
11     "entertainment_2": "Famous singer announces world tour",
12     "politics_2": "New policies aim to improve education system"
13 }
14 #input:"Elections bring significant changes to the government",output:"Headline found in category:Politics"
15 # Example usage
16 headline_to_search = input("Enter a headline to search: ")
17 found = False
18 for category, headline in news_headlines.items():
19     if headline_to_search.lower() == headline.lower():
20         print(f"Headline found in category: [{category.capitalize()}]")
21         found = True
22         break
23 if not found:
24     print("Headline not found")
25
```

Output:

```
Enter a headline to search: Elections bring significant changes to the government
Headline found in category: Politics
```

4. Few-shot Prompting

Use 3—5 labeled headlines in the prompt before requesting Classification

PROMPT: #write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment,politics by using only

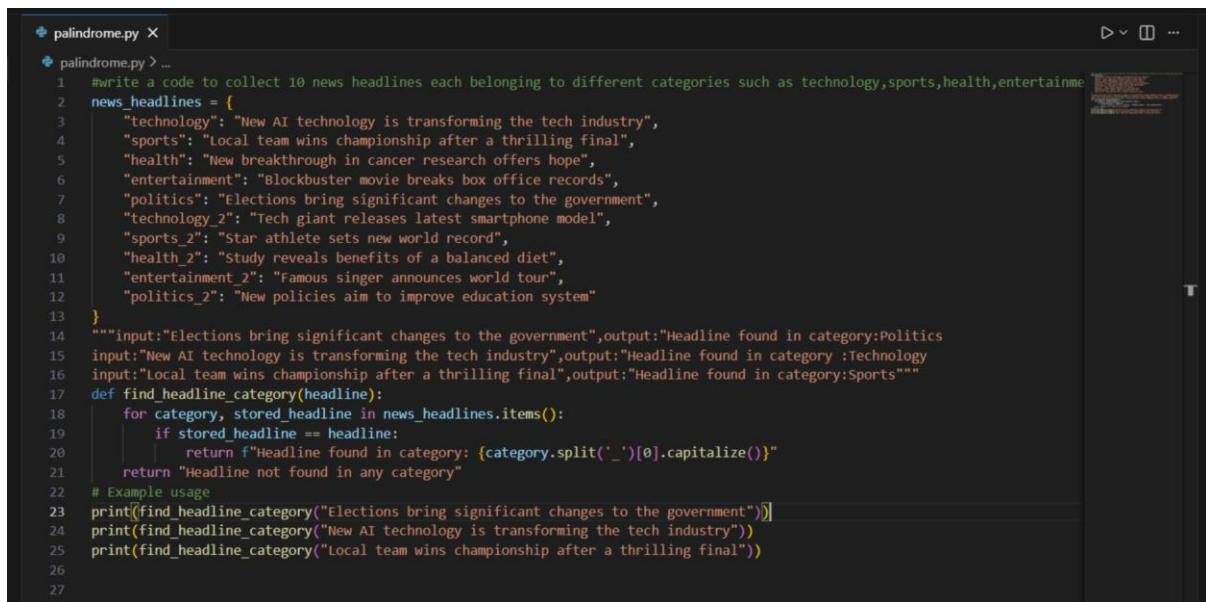
this four categories dont use other categories and store them in a dictionary where the key is category and value is headline

input:"Elections bring significant changes to the government",**output:**"Headline found in category:Politics

input:"New AI technology is transforming the tech industry",**output:**"Headline found in category :Technology

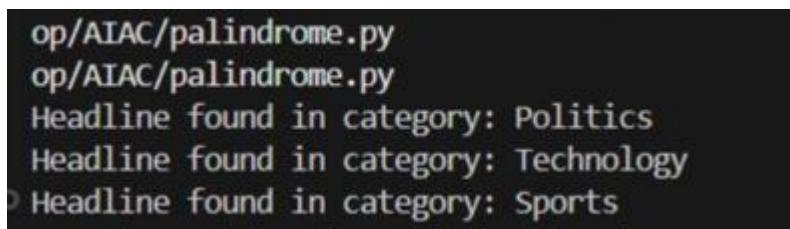
input:"Local team wins championship after a thrilling final",**output:**"Headline found in category:Sports"""

CODE:



```
palindrome.py x
palindrome.py > ...
1  #write a code to collect 10 news headlines each belonging to different categories such as technology,sports,health,entertainment
2 news_headlines = {
3     "technology": "New AI technology is transforming the tech industry",
4     "sports": "Local team wins championship after a thrilling final",
5     "health": "New breakthrough in cancer research offers hope",
6     "entertainment": "Blockbuster movie breaks box office records",
7     "politics": "Elections bring significant changes to the government",
8     "technology_2": "Tech giant releases latest smartphone model",
9     "sports_2": "Star athlete sets new world record",
10    "health_2": "Study reveals benefits of a balanced diet",
11    "entertainment_2": "Famous singer announces world tour",
12    "politics_2": "New policies aim to improve education system"
13 }
14 """input:"Elections bring significant changes to the government",output:"Headline found in category:Politics
15 input:"New AI technology is transforming the tech industry",output:"Headline found in category :Technology
16 input:"Local team wins championship after a thrilling final",output:"Headline found in category:Sports"""
17 def find_headline_category(headline):
18     for category, stored_headline in news_headlines.items():
19         if stored_headline == headline:
20             return f"Headline found in category: {category.split('_')[0].capitalize()}"
21     return "Headline not found in any category"
22 # Example usage
23 print(find_headline_category("Elections bring significant changes to the government"))
24 print(find_headline_category("New AI technology is transforming the tech industry"))
25 print(find_headline_category("Local team wins championship after a thrilling final"))
26
27
```

Output:



```
op/AIAC/palindrome.py
op/AIAC/palindrome.py
Headline found in category: Politics
Headline found in category: Technology
Headline found in category: Sports
```

5.Evaluation

Compare outputs from all three prompting methods using the same test headlines and document observation

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction, but the output may be incorrect ..We will not provide any examples here.

One-shot:

We will use only one input for example in one-shot. It improves accuracy, as the AI understands the task better from a single labelled example.

Few-shot:

We will use more than one inputs for examples in few- shot. It gives the best results. The Ai shows hogher consistency and correctly classifies most headlines.

As the number of examples increases, the AI's understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.

- Customer Email Classification

Prepare five short sample emails, each belonging to one of the above categories.

1. Write a zero-shot prompt to classify a given email into one of the categories without providing any examples.

Prompt:

```
#write a code to collect 5 shol sample emails each belonging to different categories like billing,technical support,feedback and others by using only this four categories dont use other categories and store them in a dictionary where the key is category and value is email give examples
```

```
# write a code to check whethe the give email is present inwhich category or not if present return the category else return email not found in any category
```

Code:

```
lab4.py > ...
1  #write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
2  sample_emails = {
3      "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",
4
5      "technical_support": "The application crashes every time I try to log in. Please assist.",
6
7      "feedback": "The new update is very user-friendly and much faster than before.",
8
9      "others": "I would like to know your customer support working hours.",
10
11     "billing_2": "I did not receive my invoice for last month. Kindly send it again."
12 }
13 # write a code to check whether the given email is present in which category or not if present return the category else return em
14 def find_email_category(email):
15     for category, stored_email in sample_emails.items():
16         if stored_email == email:
17             return f"Email found in category: {category.split('_')[0].capitalize()}"
18     return "Email not found in any category"
19 # Example usage
20 print(find_email_category("I was charged twice for my monthly subscription. Please help me get a refund."))
21 print(find_email_category("The application crashes every time I try to log in. Please assist."))
22
23 print(find_email_category("The new update is very user-friendly and much faster than before."))
24 #write a code to collect 10 news headlines each belonging to different categories such as technology,s
25
```

Output:

```
op/AIAC/lab4.py
Email found in category: Billing
Email found in category: Technical
Email found in category: Feedback
```

Write a one-shot prompt by including one labeled email example and ask the model to classify a new email.

Prompt: #write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback and others by using only this four categories dont use other categories and store them in a dictionary where the key is category and value is email give examples

#Input: "The application crashes every time I try to log in. Please assist.", output ^{s/}technical_support"

Code and Output:

The screenshot shows a code editor window with a dark theme. On the left, there's a sidebar with icons for search, replace, and other file operations. The main area displays a Python script named `lab4.py`. The code defines a dictionary `sample_emails` containing five entries: `"billing"`, `"technical_support"`, `"feedback"`, `"others"`, and `"billing_2"`. Each entry contains a sample email string. A function `categorize_email` takes an email as input and returns its category. An example usage is shown at the bottom. The terminal below the editor shows the command `python lab4.py` being run, followed by the output: "Email found in category: Feedback". The status bar at the bottom indicates the file is 3.13.0, the language is Python, and the current date and time are 12-01-2026.

```
1 #write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
2
3 sample_emails = {
4     "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",
5     "technical_support": "The application crashes every time I try to log in. Please assist.",
6     "feedback": "The new update is very user-friendly and much faster than before.",
7     "others": "I would like to know your customer support working hours.",
8     "billing_2": "I did not receive my invoice for last month. Kindly send it again."
9 }
10
11 #Input: "The application crashes every time I try to log in. Please assist.",output:"technical_support"
12 def categorize_email(email):
13     for category, sample_email in sample_emails.items():
14         if email == sample_email:
15             return category
16     return "Category not found"
17
18 # Example usage
19 input_email = "The application crashes every time I try to log in. Please assist."
20 output_category = categorize_email(input_email)
21 print(f'Input: "{input_email}", Output: "{output_category}"')
22
23 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\thota\OneDrive\Desktop\AIAC & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop\AIAC\lab4.py
Email found in category: Feedback
● PS C:\Users\thota\OneDrive\Desktop\AIAC & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop\AIAC\lab4.py
Input: "The application crashes every time I try to log in. Please assist.", Output: "technical_support"
○ PS C:\Users\thota\OneDrive\Desktop\AIAC

23C Mostly cloudy

In 23, Col 5 Spaces: 4 UFT-8 CR/LF () Python 3.13.0 Go Live ENG IN 21:17 12-01-2026

Write a few-shot prompt by including two or three labelled email examples and ask the model to classify a new email.

Prompt: #write a code to collect 5 shol sample emails each belonging to different categories like billling,technical suppo1,feedback and others by using only this four categories dont use other categories and store them in a dictionary where the key is category and value is email give examples

Code & Output:

The screenshot shows a code editor with a Python script named `lab4.py`. The code defines a dictionary of sample emails categorized by type (billing, technical support, feedback, others) and a function to categorize new emails based on their content. The terminal below shows the execution of the script and its output, which includes a syntax error at line 15 and the resulting classification of the input email.

```
1 #write a code to collect 5 short sample emails each belonging to different categories like billing,technical support,feedback
2 sample_emails = {
3     "billing": "I was charged twice for my monthly subscription. Please help me get a refund.",
4
5     "technical_support": "The application crashes every time I try to log in. Please assist.",
6
7     "feedback": "The new update is very user-friendly and much faster than before.",
8
9     "others": "I would like to know your customer support working hours.",
10
11     "billing_2": "I did not receive my invoice for last month. Kindly send it again."
12 }
13 """Input:"I was charged twice for my monthly subscription. Please help me get a refund.",output:"Email found in billing"
14 Input:"The application crashes every time I try to log in. Please assist.",output:"Email found in technical_support"
15 Input:"I would like to know your customer support working hours.",output:"Email found in others"""
16 def categorize_email(email):
17     for category, sample in sample_emails.items():
18         if email == sample:
19             return f"Email found in {category}"
20     return "Category not found"
21 # Example usage
22 email_to_categorize = "The application crashes every time I try to log in. Please assist."
23 result = categorize_email(email_to_categorize)
24 print(result) # Output: Email found in technical_support
25     "technical_support_2": "My internet connection drops frequently. Can you help me fix it?"}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop\AIAC\lab4.py
SyntaxError: unterminated string literal (detected at line 15)
PS C:\Users\thota\OneDrive\Desktop\AIAC> & C:/Users/thota/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thota/OneDrive/Desktop\AIAC\lab4.py
Email found in technical_support
PS C:\Users\thota\OneDrive\Desktop\AIAC>

+ v ... | ⚙ X

powershell
Python
powershell
Python
Python
Python
Python

In 25 Col 5 Spaces:4 UTF-8 CR/LF [] Python

Compare the outputs obtained using zero- shot, one-shot, and few-shot prompting techniques and briefly comment on their effectiveness

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction, but the output may be incorrect ..We will not provide any examples here.

One-shot:

We will use only one input for example in one-shot .It improves accuracy, as the AI understands the task better from a single labelled example.

Few-shot:

We will use more than one inputs for examples in few- shot.

It gives the best results .The Ai shows higher consistency and correctly classifies most headlines.

As the number of examples increases, the AI's understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.

Zero-shot:

We will not use any input examples in zero-shot .It will work with minimal instruction, but the output may be incorrect ..We will not provide any examples here.

One-shot:

We will use only one input for example in one-shot. It improves accuracy, as the AI understands the task better from a single labelled example.

Few-shot:

We will use more than one inputs for examples in few- shot.

It gives the best results. The Ai shows hogher consistency and correctly classifies most headlines.

As the number of examples increases, the AI's understanding of categories becomes clearer.

Therefore, few-shot prompting is the most effective method for news headline classification without training a model.