DSC670-JyotiDave-Week4-Exercise

October 5, 2025

1 DSC670 - Week4 - Exercise - Advanced Promt Engineering

1.1 Problem #1: Math and Extraction with Zero and One-Shot Prompting

Using the following email, build two prompts: first, create a zero-shot prompt to extract all the information from the email as JSON. Then, create a one-shot prompt to extract the information as JSON and run the extract on the 2nd email (this will be one prompt with one-shot prompting): Hey Jim, I'd like to order some new shoes. Please ship the following: 1. Nike Air Jordan I - 2 pair 2. Converse All-Star - 10 pair 3. New Balance 990 - 1 pair 4. Nike Zoom Fly 5 men's red - 2 pair Please provide sub-totals and grand total cost. Thanks. Lance Gentry 123 Main St. Chelsea, MI 48109 248-229-2229 Extract: shoe_agent_name (this is Jim) request_type (this is "Order") shoe_brand, shoe_model, shoe_quantity customer_name customer_street customer_city customer_state customer_zip customer_phone In addition, you'll assign a shoe_price for each shoe, calculate the shoe_subtotal for each shoe, and then the grand_total. Hey Michelle, I'd like to order some new shoes. Please ship the following: 1. Nike Air Jordan I - 1 pair 2. Converse All-Star - 20 pair 3. New Balance 990 - 2 pair 4. Nike Zoom Fly 5 men's red - 5 pair Please provide sub-totals and grand total cost. Thanks. Artis Gilmore 723 Lexington Blvd. New York, NY 10001 (503) 484-1029 Use one or more prompts to accomplish this extraction. You're resulting JSON should look something like this: Week 4 Exercise Output Example.txt

[4]:

```
# Suppress all warnings
import warnings
warnings.filterwarnings("ignore")
```

[14]:

```
# Step 1  Create an OpenAI client with API key

## Load required libraries
from openai import OpenAI
import json
import os
from dotenv import load_dotenv

#Load variables from .env file into environment
load_dotenv()

#Create an OpenAI client with API key stored in env file
client = OpenAI(
    # Load the api key securely from env file.
```

```python
    api_key=os.getenv("OPENAI_API_KEY")
)

# Step 2: Create a proper prompt to send it to LLM for json extraction
zero_shot_prompt = f"""
Extract the following information from this email in JSON format:

Fields to extract:
- shoe_agent_name
- request_type
- shoes (with brand, model, quantity, price, subtotal)
- grand_total
- customer_name
- customer_street
- customer_city
- customer_state
- customer_zip
- customer_phone

Email:
Hey Jim,
I'd like to order some new shoes. Please ship the following:
1. Nike Air Jordan I - 2 pair
2. Converse All-Star - 10 pair
3. New Balance 990 - 1 pair
4. Nike Zoom Fly 5 men's red - 2 pair
Please provide sub-totals and grand total cost.
Thanks.
Lance Gentry
123 Main St.
Chelsea, MI 48109
248-229-2229
"""

response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": zero_shot_prompt}],
    temperature=0
)

zero_output = response.choices[0].message.content
print("Zero-shot JSON Output:\n", zero_output)
```

~~~
Zero-shot JSON Output:
 ```json
{
  "shoe_agent_name": "Jim",
  "request_type": "order",
~~~

~~~
  "shoes": [
    {
      "brand": "Nike",
      "model": "Air Jordan I",
      "quantity": 2,
      "price": null,
      "subtotal": null
    },
    {
      "brand": "Converse",
      "model": "All-Star",
      "quantity": 10,
      "price": null,
      "subtotal": null
    },
    {
      "brand": "New Balance",
      "model": "990",
      "quantity": 1,
      "price": null,
      "subtotal": null
    },
    {
      "brand": "Nike",
      "model": "Zoom Fly 5 men's red",
      "quantity": 2,
      "price": null,
      "subtotal": null
    }
  ],
  "grand_total": null,
  "customer_name": "Lance Gentry",
  "customer_street": "123 Main St.",
  "customer_city": "Chelsea",
  "customer_state": "MI",
  "customer_zip": "48109",
  "customer_phone": "248-229-2229"
}
```
~~~

[18]:

```python
# One-shot prompt with desired JSON format
one_shot_prompt = """
Example Extraction:

Email:
Hey Jim,
I'd like to order some new shoes. Please ship the following:
```

```
1. Nike Air Jordan I - 2 pair
2. Converse All-Star - 10 pair
3. New Balance 990 - 1 pair
4. Nike Zoom Fly 5 men's red - 2 pair
Please provide sub-totals and grand total cost.
Thanks.
Lance Gentry
123 Main St.
Chelsea, MI 48109
248-229-2229

JSON Extraction (desired format):
{
'shoe_agent_name': 'Jim',
'request_type': 'Order',
'customer_name': 'Lance Gentry',
'customer_street': '123 Main St.',
'customer_city': 'Chelsea',
'customer_state': 'MI',
'customer_zip': 48109,
'customer_phone': '248-229-2229',
'items': [
{'shoe_brand': 'Nike', 'shoe_model': 'Air Jordan I', 'shoe_quantity': 2,␣
 ↪'shoe_price': 150, 'shoe_subtotal': 300},
{'shoe_brand': 'Converse', 'shoe_model': 'All-Star', 'shoe_quantity': 10,␣
 ↪'shoe_price': 60, 'shoe_subtotal': 600},
{'shoe_brand': 'New Balance', 'shoe_model': '990', 'shoe_quantity': 1,␣
 ↪'shoe_price': 180, 'shoe_subtotal': 180},
{'shoe_brand': 'Nike', 'shoe_model': "Zoom Fly 5 men's red", 'shoe_quantity':␣
 ↪2, 'shoe_price': 140, 'shoe_subtotal': 280}
],
'grand_total': 1360
}

Now extract the same JSON format from this email:

Hey Michelle,
I'd like to order some new shoes. Please ship the following:
1. Nike Air Jordan I - 1 pair
2. Converse All-Star - 20 pair
3. New Balance 990 - 2 pair
4. Nike Zoom Fly 5 men's red - 5 pair
Please provide sub-totals and grand total cost.
Thanks.
Artis Gilmore
723 Lexington Blvd.
New York, NY 10001
```

```python
(503) 484-1029
"""

# Call OpenAI API (one-shot)
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": one_shot_prompt}],
    temperature=0
)

# Extract and print the JSON output
json_output = response.choices[0].message.content
print("JSON Extraction Output:\n", json_output)
```

~~~
JSON Extraction Output:
 ```json
{
  "shoe_agent_name": "Michelle",
  "request_type": "Order",
  "customer_name": "Artis Gilmore",
  "customer_street": "723 Lexington Blvd.",
  "customer_city": "New York",
  "customer_state": "NY",
  "customer_zip": 10001,
  "customer_phone": "(503) 484-1029",
  "items": [
    {
      "shoe_brand": "Nike",
      "shoe_model": "Air Jordan I",
      "shoe_quantity": 1,
      "shoe_price": 150,
      "shoe_subtotal": 150
    },
    {
      "shoe_brand": "Converse",
      "shoe_model": "All-Star",
      "shoe_quantity": 20,
      "shoe_price": 60,
      "shoe_subtotal": 1200
    },
    {
      "shoe_brand": "New Balance",
      "shoe_model": "990",
      "shoe_quantity": 2,
      "shoe_price": 180,
      "shoe_subtotal": 360
    },
    {
~~~

```
      "shoe_brand": "Nike",
      "shoe_model": "Zoom Fly 5 men's red",
      "shoe_quantity": 5,
      "shoe_price": 140,
      "shoe_subtotal": 700
    }
  ],
  "grand_total": 2410
}
```

1.2 Similarities and differences between Zero-Shot and One-Shot Prompting

Two common approaches—zero-shot prompting and one-shot prompting—were used to extract data from customer emails describing shoe orders. This paper discusses the similarities and differences between the replies generated from each approach, highlighting how each method influences the accuracy, consistency, and quality of the responses.

In both approaches, the AI successfully understood the context of the emails and extracted key entities such as customer name, address, phone number, and product details. The zero-shot prompt provided clear instructions and an expected JSON format, allowing the model to follow a defined structure. The resulting response captured all necessary data fields—agent name, order type, and product list—demonstrating the model's strong capability for following direct instructions. Similarly, the one-shot prompt, which included an example of a previous input and output pair, produced a nearly identical structured response for the second email. This consistency shows that generative AI is reliable when working with well-defined prompts and predictable data patterns.

However, key differences exist in the level of precision and consistency between the two approaches. The zero-shot response depended entirely on the model's internal understanding of the given instructions, sometimes resulting in slight variations in data formatting or missing fields. The one-shot response, by contrast, benefited from the example provided in the prompt. By learning from that example, the AI maintained more consistent formatting and applied correct pricing and calculations for the subtotals and grand total. This difference demonstrates how contextual learning through examples enhances model accuracy and reduces ambiguity.

Overall, both prompting methods proved effective, but the one-shot technique delivered greater consistency and accuracy for structured data extraction. This reflects a key principle in prompt engineering: adding examples improves clarity and enables models to generalize instructions more effectively. As generative AI continues to evolve, the strategic use of few-shot and one-shot examples will remain an important tool for improving the precision and dependability of AI-driven data processing tasks.

1.3 Problem #2: Chat Completion for Chain of Thought

Q: There were nine computers in the server room. Five more computers were installed each day, from Monday to Thursday. How many computers are now in the server room?

A: There are 4 days from Monday to Thursday. 5 computers were added each day. That means in total 4 * 5 = 20 computers were added. There were 9 computers initially, so now there are 9 + 20 = 29 computers. The answer is 29.

Q: Michael had 58 golf balls. On Tuesday, he lost 23 golf balls. On Wednesday, he lost 2 more. How many golf balls did he have at the end of Wednesday?

A: Michael initially had 58 balls. He lost 23 on Tuesday, so after that he has 58 - 23 = 35 balls. On Wednesday he lost 2 more so now he has 35 - 2 = 33 balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?

A: She bought 5 bagels for $3 each. This means she spent $15. She has $8 left.

Q: When I was 6 my sister was half my age. Now I'm 70 how old is my sister?

[29]:
```
# Prepare prompt
problems = [
    "There were nine computers in the server room. Five more computers were␣
  ↪installed each day, from Monday to Thursday. How many computers are now in␣
  ↪the server room?",
    "Michael had 58 golf balls. On Tuesday, he lost 23 golf balls. On␣
  ↪Wednesday, he lost 2 more. How many golf balls did he have at the end of␣
  ↪Wednesday?",
    "Olivia has $23. She bought five bagels for $3 each. How much money does␣
  ↪she have left?",
    "When I was 6 my sister was half my age. Now I'm 70 how old is my sister?"
]

# Loop through problems and call OpenAI ChatCompletion
for problem in problems:
    prompt = f""" Solve the following word problem step by step and provide the␣
  ↪final answer at the end.

Problem:
{problem}
"""
    response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": prompt}],
    temperature=0
)
    answer = response.choices[0].message.content

    # Print the problem
    print(f"Problem: {problem}")

    # Print the problem. After printing the answer, it prints a line of 50␣
  ↪dashes to separate it from the next output.
    print(f"Answer:\n{answer}\n{'-'*50}\n")
```

```
Problem: There were nine computers in the server room. Five more computers were
installed each day, from Monday to Thursday. How many computers are now in the
server room?
```

```
Answer:
To solve the problem step by step, we will follow these steps:

1. **Identify the initial number of computers**:
   - There were 9 computers in the server room initially.

2. **Determine the number of days computers were installed**:
   - Computers were installed from Monday to Thursday, which is a total of 4
days.

3. **Calculate the number of computers installed each day**:
   - According to the problem, 5 more computers were installed each day.

4. **Calculate the total number of computers installed over the 4 days**:
   - Total computers installed = Number of computers installed each day × Number
of days
   - Total computers installed = 5 computers/day × 4 days = 20 computers

5. **Calculate the total number of computers now in the server room**:
   - Total computers now = Initial number of computers + Total computers
installed
   - Total computers now = 9 computers + 20 computers = 29 computers

Thus, the final answer is that there are **29 computers** now in the server
room.
--------------------------------------------------

Problem: Michael had 58 golf balls. On Tuesday, he lost 23 golf balls. On
Wednesday, he lost 2 more. How many golf balls did he have at the end of
Wednesday?
Answer:
To solve the problem step by step, we will start with the initial number of golf
balls Michael had and then subtract the number of golf balls he lost on Tuesday
and Wednesday.

1. **Initial number of golf balls**: Michael starts with 58 golf balls.

2. **Golf balls lost on Tuesday**: He lost 23 golf balls on Tuesday.
   - Calculation:
     \[
     58 - 23 = 35
     \]
   - After Tuesday, Michael has 35 golf balls left.

3. **Golf balls lost on Wednesday**: He lost 2 more golf balls on Wednesday.
   - Calculation:
     \[
     35 - 2 = 33
     \]
   - After Wednesday, Michael has 33 golf balls left.

4. **Final answer**: At the end of Wednesday, Michael has 33 golf balls.

Thus, the final answer is **33 golf balls**.
--------------------------------------------------

Problem: Olivia has $23. She bought five bagels for $3 each. How much money does
she have left?
Answer:
To solve the problem step by step, we will follow these steps:

1. **Determine the total cost of the bagels**:
   Olivia bought 5 bagels, and each bagel costs $3.
   We can calculate the total cost by multiplying the number of bagels by the
cost per bagel:
   \[
   \text{Total cost of bagels} = \text{Number of bagels} \times \text{Cost per
bagel} = 5 \times 3 = 15
   \]

2. **Calculate how much money Olivia has left**:
   Olivia started with $23 and spent $15 on bagels. We can find out how much
money she has left by subtracting the total cost of the bagels from her initial
amount:
   \[
   \text{Money left} = \text{Initial amount} - \text{Total cost of bagels} = 23
- 15 = 8
   \]

3. **Final answer**:
   After buying the bagels, Olivia has $8 left.

Thus, the final answer is:
\[
\boxed{8}
\]
--------------------------------------------------

Problem: When I was 6 my sister was half my age. Now I'm 70 how old is my
sister?
Answer:
Let's break down the problem step by step.

1. **Identify the ages when you were 6**:
   - You were 6 years old.
   - Your sister was half your age, which means she was 6 / 2 = 3 years old.
```

```
2. **Calculate the age difference**:
   - The age difference between you and your sister is 6 - 3 = 3 years. This
means you are always 3 years older than your sister.

3. **Determine your current age**:
   - You are now 70 years old.

4. **Calculate your sister's current age**:
   - Since you are 3 years older than your sister, we subtract the age
difference from your current age:
   - Your sister's age = 70 - 3 = 67 years old.

Therefore, your sister is 67 years old now.

**Final Answer**: 67 years old.
--------------------------------------------------
```

1.4 Analysis

The answers show clear step-by-step thinking, with each calculation explained.

Each solution shows the steps as well as the final answer, making it easy to understand and check.

The problems include simple math and logic about age, and the AI solved all of them correctly