

Project Title: “Prompt Engineering Framework for a Smart Virtual Assistant”

Objective:

- Chain-of-thought reasoning
 - Few-shot learning
 - JSON/structured output
 - Role-playing
 - Prompt templating
-

Project Modules:

1. Prompt Template Engine

- Create reusable prompt templates using Jinja2.
- Templates include variables like `{user_input}`, `{context}`, `{task_type}`

Deliverables:

- At least 5 prompt templates (e.g., Q&A, summarization, translation, role-play, JSON schema formatting)

Model Used: Qwen3:0.6b

```
templates = {  
    "summarization": ""Summarize the following article in bullet points.  
Article: {{ article_text }}"
```

```
Summary:""",

    "translation": """Translate the text from {{ original_language }} to {{
translated_language }}: {{ user_input }}""",

    "json_format": """Format the following text into JSON schema:
{{ user_input }}""",

    "expert_qna": """Answer the following question as an expert in {{ context }}
domain. Make the answer concise.
Question: {{ user_input }}
Answer:""",

    "roleplay_director": """You are the creative director of a film studio.
Answer the following question like a professional creative director: {{
user_input }}""",

    "general_qna": """Answer the following general knowledge question concisely.
Question: {{ user_input }}
Answer:"""
}
```

2. Few-Shot Prompting for Classification

- Implement few-shot prompts for sentiment analysis and intent classification
- Dynamically inject examples into prompts
- Compare performance vs. zero-shot mode

Deliverables:

- Classification results using GPT-2 or GPT-Neo
- Performance summary: accuracy with 0-shot vs few-shot

Model Used: Phi

Analysis:

The model was tested on zero-shot and few-shot prompts.

Zero-shot prompt:

```
"""Classify the sentiment of this sentence as Positive, Negative, or Neutral:
Sentence: "{text}"
Sentiment:"""
```

Few-shot prompt:

```
"""Classify the sentiment of the following sentences:

Example 1:
Sentence: "The service was terrible and slow."
Sentiment: Negative

Example 2:
Sentence: "Everything was perfect and I enjoyed it."
Sentiment: Positive

Example 3:
Sentence: "It's just another average day."
Sentiment: Neutral

Example 4:
Sentence: "Well, that was the best waste of two hours ever."
Sentiment: Negative

Example 5:
Sentence: "I expected more, but it was decent."
Sentiment: Neutral

Example 6:
Sentence: "Oh great, another masterpiece."
Sentiment: Negative

Now classify:
Sentence: "{text}"
Sentiment:"""
```

Six sample sentences were assigned sentiment labels and then passed into each prompt to check which prompt the model performs better on:

```
dataset = [
    {"text": "The service was terrible and slow.", "label": "Negative"},
    {"text": "It's just another average day.", "label": "Neutral"},
```

```
{ "text": "Well, that was the best waste of two hours ever.", "label":  
"Negative"},  
  { "text": "I expected more, but it was decent.", "label": "Neutral"},  
  { "text": "Thanks a lot for ruining my day.", "label": "Negative"},  
  { "text": "Absolutely stunning visuals, but a painfully slow plot.", "label":  
"Neutral"},  
  { "text": "It's fine.", "label": "Neutral"}  
]
```

Out of the six sentences, five were classified correctly by the model for each prompt i.e the model's response matched the assigned label. For the remaining one sentence though, the model classified the sentiment correctly for the few-shot prompt only:

```
-----  
Example 6: Absolutely stunning visuals, but a painfully slow plot.  
Expected Sentiment : Neutral  
Zero-Shot Response : Negative ❌  
Few-Shot Response  : Neutral ✅  
-----
```

The results showed that the model performed better on few-shot prompt. The accuracy for zero-shot was 85.71%, while for few-shot prompt, it was 100%, representing the accuracy and importance of few-shot prompts as a prompt engineering technique to help the model produce accurate results.

```
--- Accuracy Summary ---  
Zero-Shot Accuracy: 85.71%  
Few-Shot Accuracy : 100.00%
```

3. Chain-of-Thought Reasoning Module

- Build prompts to solve logical or math problems with step-by-step reasoning
- Optional: Add custom reasoning trigger like "Let's think step by step."

Deliverables:

- CoT prompt outputs for 5 different logical/math tasks
- Evaluation of correctness (can be done manually)

Model Used: Phi

Prompt 1: Multiplication Problem

```
"""
Solve the following math problem step by step:
Q: 325 * 5 = ?
A: First, break down 325 into chunks : 300 + 20 + 5. Then, multiply each by 5:
(300 x 5) + (20 x 5) + (5 x 5).
Finally, add the results (1500 + 100 + 25 = 1625).

Q: 112 * 4 = ?
A: """
```

Output:

```
Model Response:
First, break down 112 into chunks: 100 + 10 + 2. Then multiply each by 4: (100 x 4) + (10 x 4) + (2 x 4)
.
The result is (400 + 40 + 8 = 448).
```

Evaluation: Upon checking by calculator, 112 multiplied by 4 does indeed give 448. Addition of intermediary steps is also checked and found to be correct.

Prompt 2: Next Number in Sequence

```
"""
Q: What is the next number in the sequence: 2, 4, 8, 16, ?
A: Observe the pattern: each number is multiplied by 2.
2 x 2 = 4, 4 x 2 = 8, 8 x 2 = 16.
So, 16 x 2 = 32.

Q: What is the next number in the sequence: 3, 6, 12, 24, ?
A: """
```

Output:

Model Response:

This is also an interesting pattern! Each number is doubled from the previous one.
 $3 \times 2 = 6$,
 $6 \times 2 = 12$,
 $12 \times 2 = 24$.
So, following this pattern, the next number should be $24 \times 2 = 48$.

Evaluation: Upon checking by calculator, 24 multiplied by 2 does indeed give 48. Rest of the sequence is also correct.

Prompt 3: Ratio Problem

```
""  
Q: If 5 pencils cost $15, how much do 3 pencils cost?  
A: Follow a step-by-step approach. First, find the cost of 1 pencil:  $\$15 \div 5 = \$3$   
per pencil.  
Now multiply by 3:  $\$3 \times 3 = \$9$ .  
  
Q: 6 apples cost $24. How much will 10 apples cost?  
A:""
```

Output:

Model Response:
To find out the cost of one apple, divide \$24 by 6. This gives us \$4 per apple. Next, multiply this price by 10 to get the total for 10 apples: $\$4 \times 10 = \40 . Therefore, 10 apples will cost \$40.

Evaluation: Upon checking by calculator, 24 divided by 6 does indeed give 4. Rest of the logic is basic algebra checked manually. It is correct.

Prompt 4: Adding Large Numbers

```
""  
Q:  $144 + 256 = ?$   
A: First, break down the numbers:  
 $144 = 100 + 40 + 4$   
 $256 = 200 + 50 + 6$   
  
Now add corresponding parts:  
 $100 + 200 = 300$   
 $40 + 50 = 90$   
 $4 + 6 = 10$ 
```

Finally, add all results:

$$300 + 90 + 10 = 400$$

Q: $345 + 102 = ?$

A: First, break down the numbers:

$$345 = 300 + 40 + 5$$

$$102 = 100 + 0 + 2$$

Now add corresponding parts:

$$300 + 100 = 400$$

$$40 + 0 = 40$$

$$5 + 2 = 7$$

Finally, add all results:

$$400 + 40 + 7 = 447$$

Q: $256 + 123 = ?$

A: ""

Output:

Model Response:

First, break down the numbers:

$$256 = 200 + 50 + 6$$

$$123 = 100 + 20 + 3$$

Now add corresponding parts:

$$200 + 100 = 300$$

$$50 + 20 = 70$$

$$6 + 3 = 9$$

Finally, add all results:

$$300 + 70 + 9 = 379$$

Evaluation: Upon checking by calculator, 256 plus 123 does indeed give 379. Addition of intermediary steps is also checked and found to be correct.

Prompt 5: Maths and Logics Problem

```
"""
Respond with step-by-step logic and a short final answer only. Do not add extra
commentary or characters.

Q: Tom is twice as old as Jerry. If Jerry is 5 years old, how old is Tom?
A: First, note that Tom is 2 times Jerry's age.
Now multiply:  $5 \times 2 = 10$  years old.
Now answer the following using the same steps:

Q: A bookshelf has 4 shelves. If each shelf holds 7 books, how many books are
there in total?
A: """
```

Output:

```
Model Response:
    The number of books on each shelf is 7.
To find out how many books in total, you need to multiply the number of shelves by the number of books per
shelf:  $4 \times 7 = 28$ . So, there are 28 books in total.
```

Evaluation: Upon checking by calculator, 4 multiplied by 7 does indeed give 28. Logic is applied correctly.

4. JSON Mode and Structured Output Generator

- Implement prompts that extract data in clean JSON
- Optionally define a mock schema (e.g., user profile, product info)
- Parse model output to check if valid JSON (use Python's `json` module)

Deliverables:

- At least 3 JSON-structured outputs
- A function to validate model response against expected schema

Model Used: Phi

JSON Structured Outputs:

User Profile:

Prompt:

```
""" You are an expert JSON Formatter. Extract a user profile in JSON format with the following fields: name, age, email, and interests. Here's the user description:
```

```
"Sarah is a 28-year-old software engineer who enjoys hiking, painting, and reading tech blogs. You can contact her at sarah.mills@example.com"
"""
```

Output:

Model Response:

```
{
  "name": "Sarah",
  "age": 28,
  "email": "sarah.mills@example.com",
  "interests": ["hiking", "painting", "reading tech blogs"]
}
```

Parsed JSON:

```
{
  "name": "Sarah",
  "age": 28,
  "email": "sarah.mills@example.com",
  "interests": [
    "hiking",
    "painting",
    "reading tech blogs"
  ]
}
```

Schema is valid.

Product Info:

Prompt:

```
"""
You are an expert JSON Format. Generate a product listing in JSON format with
fields: product_id, name, price, availability, tags (key words).

Here is the product description:

"The EcoBottle X200 (ID: EBX200) is available for $19.99. It's a reusable bottle
made for outdoor use and eco-conscious consumers."
"""
```

Output:

Model Response:

```
{
  "product_id": "EBX200",
  "name": "EcoBottle X200",
  "price": "$19.99",
  "availability": true,
  "tags": ["outdoor", "eco-conscious"]
}
```

Parsed JSON:

```
{
  "product_id": "EBX200",
  "name": "EcoBottle X200",
  "price": "$19.99",
  "availability": true,
  "tags": [
    "outdoor",
    "eco-conscious"
  ]
}
```

Schema is valid.

Book Data:

Prompt:

```
"""
You are an expert JSON Formatter. Extract book metadata from the description in
JSON format with the fields: title, author, year, genre, and summary.

Description:
"'The Midnight Library' by Matt Haig, published in 2020, explores the choices
that shape our lives. It blends science fiction with philosophical themes."
"""
```

Output:

```
Model Response:

{
  "title": "The Midnight Library",
  "author": "Matt Haig",
  "year": 2020,
  "genre": ["Science Fiction", "Philosophical"],
  "summary": "Explores the choices that shape our lives. Blends science fiction with philosophical themes."
}

Parsed JSON:
{
  "title": "The Midnight Library",
  "author": "Matt Haig",
  "year": 2020,
  "genre": [
    "Science Fiction",
    "Philosophical"
  ],
  "summary": "Explores the choices that shape our lives. Blends science fiction with philosophical themes."
}
Schema is valid.
```

Function to validate model response against expected schema:

```
def validate_schema(data, expected_fields):
    return expected_fields.issubset(data.keys())
```

5. Role-Playing Prompting Engine

- Design prompts that assign the model roles like:

- Doctor
- Lawyer
- Customer Support Agent
- Compare response styles with and without role prompts

Deliverables:

- 3 comparisons of model behavior with vs. without a role prompt
- Insights into tone, accuracy, and domain relevance

Model Used: Phi

Doctor:

Without role prompt:

Prompt:

"I've had a sore throat and mild fever for three days. Should I be concerned?"

Output:

Model's Response Without Role Play:

Doctor:

It's always better to take care of your health, especially if you have been experiencing symptoms for several days. In general, it's a good idea to see a doctor when you are having persistent symptoms that last more than a few days or get worse over time. Sore throat and mild fever can be signs of various conditions like strep throat, tonsillitis, or even the flu. It's always best to consult with a medical professional who will evaluate your specific situation and determine if any further action is needed.

With role prompt:

Prompt:

""You are an experienced medical doctor specialized in general medicine. Answer the patient concisely.
The user asks: 'I've had a sore throat and mild fever for three days. Should I be concerned?'
Doctor: ""

Output:

Model's Response With Role Play:

Doctor:

I would recommend that you make an appointment with your primary care physician or a specialist in infectious diseases to get a proper diagnosis. They can assess the severity of your symptoms, perform any necessary tests, and provide appropriate treatment if needed. It's always better to err on the side of caution when it comes to health concerns.

Analysis:

The role-play response adopts a professional, authoritative tone, directly recommending consultation with a primary care physician or infectious disease specialist while emphasizing proper diagnosis and treatment, making it more actionable and clinically precise. In contrast, the non-role-play response offers generic advice in a casual tone, listing possible conditions like strep throat or flu but lacking specific guidance or urgency. Overall, the role-play version excels in credibility, domain relevance, and structured guidance, while the non-role-play response remains accurate but less impactful for medical decision-making.

Lawyer:

Without role prompt:

Prompt:

"Can I file for custody if I don't have a stable income yet?"

Output:

Model's Response Without Role Play:

Lawyer:

Yes, it is possible to file for custody even if you do not have a stable income at this time. In order to file for custody, you will need to provide evidence of your ability to support your child financially, such as proof of employment or other sources of income. It's important to note that the court will take into consideration factors like your ability to provide for your child, your work history and your financial situation when making a decision about custody.

With role prompt:

Prompt:

""You are a professional legal advisor with expertise in family law. Answer the user concisely.

The user asks: 'Can I file for custody if I don't have a stable income yet?'

Lawyer: ""

Output:

Model's Response With Role Play:

Lawyer:

Yes, you can file for custody even if you do not currently have a stable income. The court takes into consideration various factors when determining custody arrangements, including the best interests of the child, and financial stability is just one aspect that they consider. It's important to gather evidence to show that you are committed to providing a safe and nurturing environment for your child, even if you're not currently financially secure.

Analysis:

The role-play response adopts a more professional and client-focused tone, emphasizing the "best interests of the child" and reassuring the individual that financial stability is just one factor among many. It encourages gathering evidence of a safe and nurturing environment, making it more nuanced and supportive.

The non-role-play response, while factually correct, is more transactional, focusing narrowly on financial proof (employment, income sources) and court considerations without the same level of empathy or strategic guidance.

Customer Support Agent:

Without role prompt:

Prompt:

"I received a damaged laptop in my order. What should I do?"

Output:

Model's Response Without Role Play:

Customer Support Agent:

I'm sorry to hear that you received a damaged laptop in your order. Here are some steps you can take:

1. contact the seller or e-commerce platform immediately to report the issue and provide photographic evidence of the damage.
2. follow any instructions provided by the seller or platform for returning the damaged item for replacement or refund.
3. keep a record of all communication with the seller, including dates and times of phone calls or emails.
4. check your warranty to see if it covers accidental damage or shipping issues.
5. file a complaint with the relevant consumer protection agency if the issue cannot be resolved internally.

Remember to remain polite and professional in all communications with the seller, as they may need to assist you further.

Here's an interesting puzzle that relates our conversation to a hypothetical scenario:

The user received 4 laptops from four different sellers (Seller A, Seller B, Seller C, and Seller D). Each laptop was shipped on a separate day of the week (Monday to Thursday). The users also had varying levels of satisfaction with each seller.

Here are some clues:

1. Seller A did not ship their product on Monday or Tuesday.
2. The user was most satisfied with the vendor who shipped their laptop on Wednesday.
3. Seller C's delivery arrived after the one from Seller B but before the one from Seller D.
4. The user was least satisfied with the seller whose delivery occurred on Thursday.
5. Seller A's product had been in transit for less time than the one sold by Seller C, but more time than that of Seller D.
6. The laptop shipped by Seller B arrived earlier in transit than that of Seller A.

With role prompt:

Prompt:

```
""You are a polite and helpful customer support agent for an online electronics store. Answer the customer concisely in 2-3 sentences.
The user asks: 'I received a damaged laptop in my order. What should I do?'
Customer Support Agent:""
```

Output:

Model's Response With Role Play:

Customer Support Agent:

Hello, thank you for bringing this to our attention. We apologize for any inconvenience caused by receiving a damaged item. Please contact our customer service team with your order number and photos of the damage. We will arrange for a replacement or refund as soon as possible. Thank you for choosing us!

Analysis:

The non-role-play response hallucinated by abruptly switching from customer service advice to an unrelated shipping puzzle—a clear deviation from the prompt. It was also overly verbose, listing excessive steps for a damaged laptop issue. In contrast, the role-play response stayed on-topic, concise, and brand-specific, proving more effective for customer support interactions.

6. Unified Task Router

- Implement a simple interface (CLI or Streamlit)
- Based on user selection (`task_type`: "qa", "summarization", "roleplay"), the system selects and fills the right prompt

Deliverables:

- Unified script or web app
- Clear routing logic using `if-else` or dictionary-based dispatch

<https://github.com/Fatima-Siddiq/TASK-BASED-AI-ASSISTANT>
