



Concise Video Game Data Enhancement with Google AI Studio API

Eclipse AI Data Optimization

Sultan Fahd M.B.Y

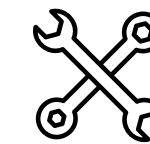
 sultanyusuf2936@gmail.com

Task and Tools



Task

- Classify game genres based on title
- Generate concise gameplay descriptions (less than 30 words)
- Determine player modes (Singleplayer, Multiplayer, Both)
- Leverage Google AI Studio and Python automation



Tools



These tools enable scalable automation, efficient data processing, and real-time interaction with Google's Gemini API through Python.

Prompt Engineering

Good Prompt

```
def call_gemini(title):  
    """Call Gemini API with combined genre, description, and player mode prompt"""  
    prompt = f"""You are an expert game analyst. Analyze the game title below and respond with a strict JSON format.
```

Clear Role Definition
(Identity)

```
GAME TITLE: {title}
```

```
RESPONSE FORMAT (JSON ONLY, NO EXPLANATION, NO MARKDOWN):
```

```
{  
  "genre": "[EXACTLY ONE from: Action, Adventure, RPG, Strategy, Simulation, Sports, Puzzle, Horror, FPS, Platformer, Fighting, Racing, MMORPG, Sandbox, Survival, BattleRoyale]",  
  "description": "[Concise, 20-30 word third-person game description, active voice, no pronouns]",  
  "player_mode": "[Singleplayer / Multiplayer / Both]"  
}
```

Specific Task Description

```
EXAMPLE:
```

```
{  
  "genre": "FPS",  
  "description": "Tactical first-person shooter emphasizing agent abilities and precise shooting mechanics across objective-based competitive multiplayer matches.",  
  "player_mode": "Multiplayer"  
}
```

Concrete Example



Breakdown Prompt

- **Role/Identity:** Gemini **acts** as an expert game analyst to ensure domain-relevant output.
- **Context:** Input is just a **game title**.
- **Prompt/Query:** Task is **concise** and directive
- **Output Format:** strict **JSON** with predefined choices
- **Validation Constraints:** genre must **match exact** list; word count; no pronouns or passive voice



Why This Is a Good Prompt?

- ✓ **Clear instructions** : No room for ambiguity in task or format.
- ✓ **Strict format with constraints** : Helps ensure structured and valid.
- ✓ **Minimizes error** : Limits output to predefined options
- ✓ **Includes positive example** : Models perform better when shown a reference
- ✓ **Token-efficient** : Avoids unnecessary verbosity and explanations.

This prompt becomes more focused, reproducible,
and easily processed

Prompt Engineering

Bad Prompt

```
def call_gemini(title):  
    """Call Gemini API with combined genre, description, and player mode prompt"""  
    prompt = f"""I have a game title called "{title}" and I need your help to better understand it.  
  
Can you please tell me:  
1. What genre does this game belong to?  
2. What is the gameplay like? Please describe this game.  
3. Does the game support singleplayer, multiplayer, or both modes?  
"""
```

Context

Specific Task Description

Breakdown Prompt

- **Role/Identity:** The model **isn't** told to act as an expert or in any specific role.
- **Context:** Input is just a **game title**.
- **Prompt/Query:** Very open-ended: asks what kind of game it is and requests a description.
- **Output Format:** No structure defined
- **Validation Constraints:** None



Why This Is a Bad Prompt?

- ✗ **No structured format** : The prompt is unstructured and ambiguous.
- ✗ **No Validation Constraints** : The output may include made-up genres, pronouns, or verbose descriptions
- ✗ **Unreliable output labels** : Limits output to predefined options
- ✗ **Token-Inefficient** : Uses conversational, which increases token usage without adding value.

This prompt provides too little guidance and too much freedom, often producing unpredictable results, excessive token usage, and responses that are unsuitable for structured data workflows.

Challenges & Solutions



Challenge 1: API Request Limit (50/day)

- **Problem:** Google AI Studio API only allows 50 requests per day on the free plan.
- **Impact:** Slowed down batch processing, unable to process all game titles at once.

Solution:

- ✓ Optimized prompt design to extract all 3 outputs (genre, description, player mode) in one request
- ✓ Limited test runs to 3–10 samples during development.
- ✓ Option to upgrade to paid plan if needed for production scale



Challenge 2: Unstructured Output in Early Attempts

- **Problem:** Outputs were verbose or inconsistent without clear formatting.
- **Impact:** Parsing failed in automation scripts.

Solution:

- ✓ Enforced strict JSON format
- ✓ Provided detailed example within the prompt

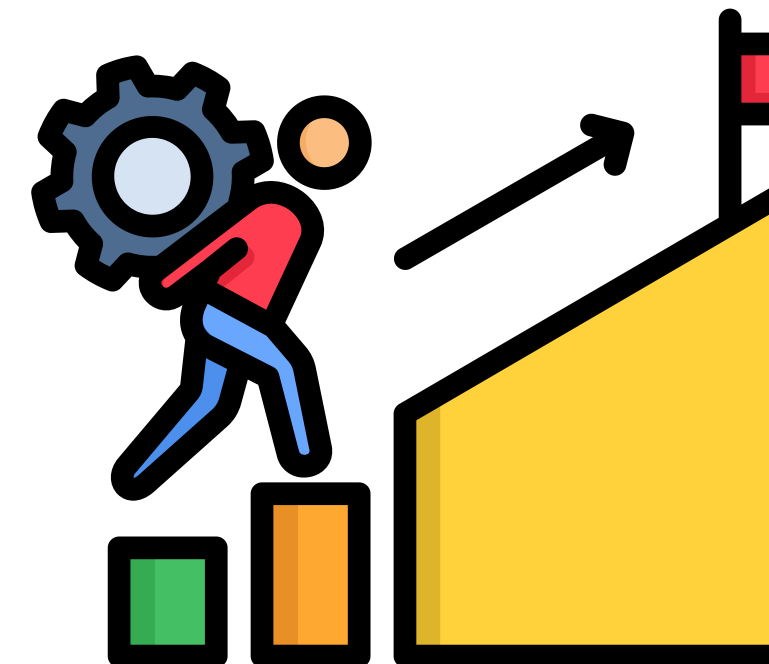


Challenge 3: Token Inefficiency in Prompting

- **Problem:** Separate prompts for genre, description, and mode increased token use.
- **Impact:** Higher latency and cost, limited daily request capacity.

Solution:

- ✓ Consolidated all tasks into a single structured prompt
- ✓ Removed unnecessary explanations and enforced concise outputs
- ✓ I used 5 account google for development prompt and code



Conclusion & Reflections

Conclusion

- A well-crafted prompt significantly affects response accuracy, clarity, and token efficiency
- Combining multiple output (genre, description, and player mode) into JSON format optimized API usage.
- Understanding free API limits early helps in planning scalable automation strategies

Reflection

This project highlight the important of prompt engineering and API integration in automating classfication task. it demonstrates how strategic prompt design directly affects output quality and processing efficiency.





Task 2: Concise Video Game Data Enhancement with Google AI Studio API

Thank You

Sultan Fahd M.B.Y

 sultanyusuf2936@gmail.com