Prompt Engineering Documentation

By: Soubhagya Srivastava

Email: soubhagyasrivastava240@gmail.com

[GitHub Repository]

File 1: agents/story.py Prompt

1. Objective of the Prompt

The primary goal of this prompt is to instruct the AI model to produce structured creative content in a strict JSON format, ensuring machine-readable output without additional commentary. This serves both creative and technical purposes — creative in generating a story, and technical in ensuring consistent parsing.

2. Prompt Structure Breakdown

The prompt is crafted in clear instruction blocks, each contributing to precision and reliability:

a) Role Assignment

`“You are a creative writer.”`

- Purpose: Establishes the model’s *persona*, influencing its tone and style.
- Benefit: Increases creativity and narrative flow while avoiding overly technical or robotic responses.

b) Task Specification

```
“ Given the following user input, produce a JSON object with exactly
three keys:
1) "short_story" - ...
2) "character_description" - ...
3) "background_description" - … ”
```

- Purpose: Defines the exact output structure required.
- Benefit: Reduces ambiguity by listing the exact keys and the type of content expected for each.
- Technique Used: Enumerated constraints — a list with numbered items ensures the model follows a fixed sequence.

---

c) Content Guidelines

Each key has inline detailed sub-instructions:

- `"short_story"` → specifies length (6–12 sentences) for control over verbosity.
- `"character_description"` → lists detailed physical and stylistic attributes.

- `"background_description"` → includes environment, time, colors, textures, mood, and style suggestions.
- Purpose: Provides rich attribute detail to avoid vague or incomplete responses.
- Benefit: Encourages the model to output comprehensive, vivid, and visually descriptive results.

---

d) Output Formatting Enforcement

```
“ Return ONLY valid JSON (no extra commentary). Make sure strings are properly escaped and the JSON can be parsed by a program. ”
```

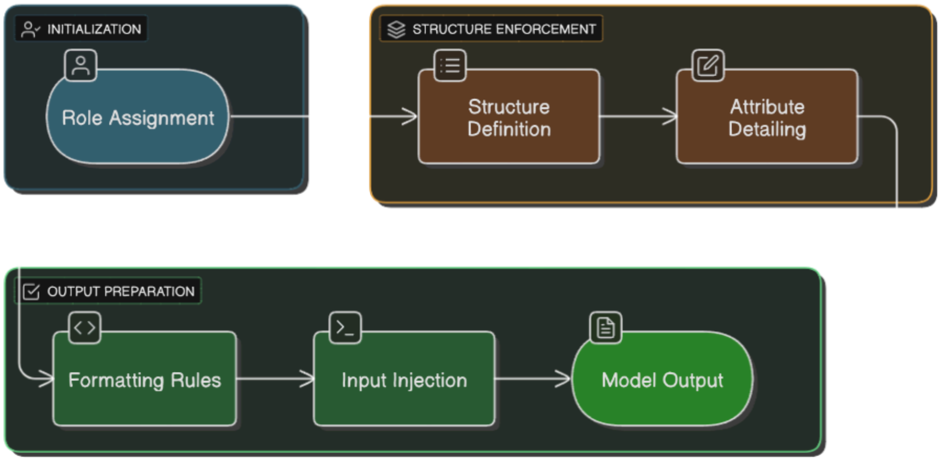
- Purpose: Forces strict adherence to machine-readable JSON format.
- Benefit: Eliminates extra text, ensuring zero parsing failures when programmatically handling responses.
- Technique Used: Explicit prohibitions (“no extra commentary”) and positive instructions (“strings properly escaped”).

---

e) Input Injection

```
“ User input:
\"\"\"{user_input}\"\"\" ”
```

- Purpose: Clearly separates dynamic user input from the main instruction set.
- Benefit: Prevents confusion between instructions and user-provided content, maintaining structural clarity.
- Technique Used: Triple-quote encapsulation to visually and logically isolate input**.**



Pictorial chart of prompt execution

File 2: agents/prompt.py

1. Objective of the Prompt

The aim is to generate a highly detailed, visually rich, and realistic image generation prompt tailored for AI image generation tools. The prompt is designed to ensure conciseness (40 words), vivid realism, and story alignment while avoiding any unnecessary explanations.

---

2. Prompt Structure Breakdown

a) Task Definition

```
Create a highly detailed and visually rich very realistic image
generation prompt.
The prompt should help an AI image generation tool visualize the
description below.
```

- Purpose: Clearly states the goal — generating a *realistic* and *visually rich* prompt.
- Benefit: Establishes the creative expectation for the model before introducing constraints.
- Technique: Specific creative directive to guide tone, style, and realism.

---

b) Output Restriction

```
Write only the final image generation prompt.
Do not include explanations, introductions, or extra text — just the
prompt itself.
```

- Purpose: Forces output to be purely the usable image prompt without extra formatting or commentary.
- Benefit: Reduces post-processing; guarantees a clean, tool-ready prompt.
- Technique: Negative instruction (“do not include…”) paired with positive directive (“just the prompt itself”).

---

c) Input Injection

```
Description: {description}.
```

- Purpose: Injects dynamic description into the prompt generation process.
- Benefit: Keeps the architecture adaptable for any context or story element.
- Technique: Variable substitution ensures flexibility for different scenes.

d) Realism & Story Alignment

```
Make sure it's realistic, vivid, and suitable according to the story
for image generation.
```

- Purpose: Ensures visual richness aligns with story tone and realism.
- Benefit: Prevents generic or fantasy-like outputs unless the description demands it.
- Technique: Stylistic constraint embedded in instruction**.**

e) Conditional Rule for Character Presence

```
If the description contains a character or person, depict them
occupying only half of the image space, else ignore.
```

- Purpose: Adds conditional logic to control composition.
- Benefit: Improves balance in generated imagery, making character inclusion deliberate and proportionate.
- Technique: Context-aware conditional instruction.

f) Word Count Constraint

```
THE PROMPT SHOULD BE WITHIN 40 WORD PARAGRAPH
```

- Purpose: Forces concise and dense descriptive writing.
- Benefit: Keeps the output short, focused, and highly optimized for image generation tools.
- Technique: Quantitative constraint to control verbosity.

File 3: img_agents/imagegen.py

1. Objective of the Negative Prompt

The negative prompt acts as a quality filter for AI image generation, telling the model what to avoid. Instead of describing what should appear, it lists undesirable elements that would harm realism, composition, or quality.

This ensures the generated image is free of common visual flaws, producing cleaner, more realistic outputs.

---

2. Negative Prompt Structure Breakdown

a) Purposeful Categorization of Unwanted Features

```
cartoon, anime, NSFW, 3d render
```

- Purpose: Avoids styles that conflict with “realistic” intent.
- Benefit: Prevents accidental generation of stylized, fictional, or unsafe content.
- Category: Style & Content Restrictions

---

b) Technical Quality Control

```
low quality, low resolution, blurry, out of focus
```

- Purpose: Eliminates resolution-related flaws.
- Benefit: Maintains sharpness and detail for realism.
- Category: Clarity & Resolution Filters

---

c) Anatomical Accuracy Enforcement

```
extra fingers, mutated hands, bad hands, missing fingers, distorted face
```

- Purpose: Corrects frequent AI generation errors with hands, faces, and anatomy.
- Benefit: Improves believability of human figures.
- Category: Anatomy Correction

---

d) Texture and Skin Realism

```
unnatural skin texture, bad anatomy, deformed, disfigured
```

- Purpose: Avoids unnatural body textures or shapes.
- Benefit: Ensures natural human appearance.
- Category: Skin & Body Refinement

---

e) Background and Environment Control

```
blurry background, text, watermark, grainy
```

- Purpose: Maintains clean, professional-looking backgrounds.
- Benefit: Prevents distractions or accidental branding.
- Category: Environment Quality Filters

---

f) Lighting & Exposure Control

```
overexposed, underexposed, unnatural lighting
```

- Purpose: Avoids lighting conditions that reduce realism.
- Benefit: Maintains balanced and natural illumination.
- Category: Lighting Quality Management

Clarity & Reliability Enhancements

(Combined for All Three Prompts)

1. **Clear Role & Context Alignment** → Assigns explicit creative or functional roles (e.g., *creative writer*, *realistic image prompt creator*) while ensuring outputs match the intended story tone and style.

2. **Strict Structure & Formatting Rules** → Enforces precise formats (e.g., JSON, plain paragraph) with clearly separated dynamic inputs, avoiding extra commentary or parsing issues.

3. **Detailed Content Guidance** → Uses enumerated attributes, descriptive checklists, and semantic guardrails to ensure completeness and relevance in every output.

4. **Adaptive Logic & Conciseness** → Applies conditional rules based on context (e.g., character placement) and explicit word limits for sharp, focused outputs.

5. **Integrated Quality Control** → Combines positive instructions (what to include) with negative prompts (what to avoid) to enhance realism and control.

6. **Technical & Visual Safeguards** → Proactively prevents common AI output flaws (e.g., watermarks, low resolution, anatomical errors, lighting issues).

7. **Reusability & Scalability** → Maintains a modular design so prompts can be reused and adapted for different scenarios without structural changes.