

ARTISAN Instruction Framework for Prompt Engineering

A Framework for Eliciting High-Quality Responses from Large Language Models

Eamonn O'Raghallaigh, PhD

Trinity Business School, Trinity College Dublin, Ireland & Digital Strategy Consultants Ltd.,
Dublin Ireland

Date Published: 9th January 2025

Introduction

The ARTISAN Instruction Framework is a systematic methodology designed to help users craft effective prompts for interacting with Large Language Models (LLMs). By adopting the principles and techniques outlined in this framework, you can significantly improve the accuracy, relevance, and overall quality of LLM outputs, unlocking their full potential for a wide range of applications.

Core Principle: The Intelligent, but Literal, Colleague

At its heart, the ARTISAN framework is built on a fundamental shift in mindset. Instead of viewing LLMs as magical oracles or advanced search engines, we must approach them as highly intelligent, but literal, colleagues. These "colleagues" possess vast knowledge and remarkable language processing abilities but require precise, well-structured instructions to apply their capabilities effectively. They interpret instructions literally and rely solely on the information provided.

The A.R.T.I.S.A.N. Acronym

The framework is organized around the acronym **A.R.T.I.S.A.N.**, representing the seven key stages involved in crafting exceptional prompts:

A - Audience & Goal Articulation

1. **Know Your Audience (The LLM):**

- **Tokenization:** LLMs process language in discrete units called tokens. Concise phrasing can improve processing efficiency.
- **Attention Mechanisms:** LLMs weigh different parts of the input differently. Emphasize crucial information through placement and formatting.

- **Training Data Influence:** Responses are shaped by the patterns and information learned during training. Awareness of this can help anticipate biases.
- **Probabilistic Nature:** Outputs are based on probabilities. Clear instructions minimize ambiguity and guide the model toward the desired outcome.

2. Define Your Goal with Precision:

- **Task Type:** Clearly specify the desired operation (e.g., summarisation, generation, translation, classification, explanation).
- **Desired Format:** Indicate the preferred output structure (e.g., paragraph, list, table, code snippet).
- **Specific Constraints:** Define limitations related to length, style, tone, perspective, and target audience.
- **Evaluation Criteria:** Consider how you will assess the quality of the response. What are the key success factors?

R - Role & Responsibility Assignment

1. **Define the LLM's Role:** Assign a specific persona or role to contextualize the task and guide the LLM's approach. Examples:
 - "You are a renowned historian specialising in..."
 - "Act as a coding tutor for beginners..."
 - "You are a creative writer tasked with..."
2. **Specify Responsibilities:** Break down the task into manageable steps or components to guide the LLM's thought process. Example:
 - Instead of: "Write a report on climate change."
 - Consider: "First, summarise the key scientific findings. Second, discuss the potential economic impacts. Finally, propose three actionable solutions."

T - Task Instruction & Clarification

1. **Use Action-Oriented Verbs:** Begin instructions with strong verbs that clearly define the desired action (e.g., "Summarise," "Analyse," "Generate," "Explain," "Compare," "Evaluate").
2. **Be Explicit and Unambiguous:** Avoid jargon, ambiguity, and implicit assumptions. State your requirements precisely.
3. **Provide Necessary Context:** Supply relevant background information, including facts, data, scenarios, the intended audience, and the purpose of the task.

4. **Clarify Constraints and Boundaries:** Specify what the LLM should *not* do or include. Examples:
 - "Do not include personal opinions."
 - "Focus only on the last decade."
 - "Keep the explanation accessible to a layperson."

I - Information Input & Examples

1. **Provide High-Quality Source Material:** If the task involves analysing or generating content based on specific information, supply clear and relevant sources (e.g., text snippets, data points, web links).
2. **Utilise Examples (Few-Shot Learning):** Demonstrate the desired output format, style, or reasoning process by providing a few illustrative examples.
 - "Here is an example of the format I need: [Example]"
 - "Based on these examples, generate three more..."
3. **Clearly Link Information to the Task:** Explicitly connect the provided information to the specific instructions. Example: "Using the provided article, summarise the main arguments."

S - Structure & Formatting Guidance

1. **Specify the Desired Output Structure:** Indicate how the information should be organised (e.g., bullet points, numbered list, essay with specific sections).
2. **Use Formatting Cues:** Employ formatting within your prompt (e.g., bolding, italics, headings, code blocks) to highlight key instructions or provide visual structure.
3. **Consider Delimiters:** Use clear delimiters (e.g., "---", "###", "```) to separate different parts of the prompt or to mark the beginning and end of input data.

A - Assessment & Refinement Loop

1. **Iterative Process:** Recognize that prompt engineering is often an iterative process.
2. **Analyse the Output:** Critically evaluate the LLM's response. Identify strengths and weaknesses in relation to your goals.
3. **Refine Your Prompt Based on Feedback:** Adjust instructions, provide more context, clarify constraints, or add examples based on the initial response.
4. **Experiment with Different Phrasing and Techniques:** Explore alternative ways of expressing your instructions to optimize results.

N - Nuance & Advanced Techniques

1. **Temperature Control:** Adjust the "temperature" parameter to influence the randomness of the output. Lower temperatures yield more deterministic responses, while higher temperatures encourage creativity.
2. **Top-P Sampling:** Similar to temperature, Top-P sampling controls the diversity of the output by considering a subset of the most probable tokens.
3. **Chain-of-Thought Prompting:** Guide the LLM through a step-by-step reasoning process for complex tasks. Example: "Let's think step by step. First, identify the key arguments. Second, analyse the evidence supporting each argument..."
4. **Prompt Decomposition:** Break down complex tasks into smaller, more manageable sub-prompts to improve accuracy and control.
5. **Negative Constraints:** Explicitly stating what *not* to do can be a powerful way to guide the LLM.
6. **Prompt Engineering Patterns:** Familiarise yourself with established techniques like "Zero-Shot," "One-Shot," and "Few-Shot" learning to leverage the LLM's capabilities effectively.

Key Takeaways

- **Precision is paramount.**
- **Structure prompts logically.**
- **Embrace iteration and refinement.**
- **Experiment and explore different techniques.**
- **Leverage examples effectively.**

Conclusion

The ARTISAN Instruction Framework provides a robust and adaptable methodology for crafting prompts that elicit high-quality responses from LLMs. By consistently applying these principles and continuously refining your approach, you can master the art of prompt engineering and unlock the full potential of these powerful AI tools.

Appendix: Framework Application in Use Case Scenarios

This section is to be used as a guide for building your own prompts, it is not part of the core framework

Creating System Instructions for a Custom LLM

System instructions are foundational directives that govern a custom LLM's behavior across all interactions. They define its core personality, constraints, and capabilities.

Applying A.R.T.I.S.A.N. to System Instructions:

A - Audience & Goal Articulation (for the LLM Developer/User):

- **Know Your Audience (the LLM and its Users):** The "audience" here is twofold: the LLM itself and the users who will interact with it.
- **Define Your Goal (for the LLM):** What is the overarching purpose of this custom LLM? Examples:
 - "To act as a knowledgeable and helpful customer service chatbot."
 - "To be a creative writing assistant focused on generating fictional narratives."

R - Role & Responsibility Assignment (for the LLM):

- **Clearly Define the LLM's Core Role:** This is the fundamental persona the LLM will embody. Examples:
 - "You are a friendly and efficient customer service representative named 'AssistBot'."
 - "You are a meticulous and objective academic researcher."
- **Specify Core Responsibilities:** What are the primary functions the LLM should perform? Examples:
 - "Answer customer inquiries accurately and promptly."
 - "Summarise research papers and identify key findings."

T - Task Instruction & Clarification (for the LLM's General Behavior):

- **Use Action-Oriented Verbs (for Consistent Behavior):** Define core behavioral guidelines. Examples:
 - "Always prioritise factual accuracy."
 - "Strive for concise and easy-to-understand language."
- **Be Explicit and Unambiguous (for Robustness):** Address potential areas of confusion. Examples:

- "When asked for personal opinions, state that you are an AI."
- "If unsure of an answer, state that you don't know and offer potential resources."
- **Provide Necessary Context (for its Domain):** Define the LLM's knowledge base and limitations.
- **Clarify Constraints and Boundaries (for Safety and Functionality):** Set limits on the LLM's actions.

I - Information Input & Examples (as Guides for Learning):

- **Provide Foundational Knowledge and Data (Training Data):** System instructions inform the selection and curation of training data.
- **Utilise Example Interaction Patterns:** Include examples of desired conversational flows within the system instructions.

S - Structure & Formatting Guidance (for Internal Consistency):

- **Specify Desired Response Structure (General Guidelines):** Outline preferred formats for different interactions.
- **Use Formatting Cues (for System Instruction Clarity):** Organize instructions logically using headings, bullet points, and clear language.

A - Assessment & Refinement Loop (for Ongoing Development):

- **Establish Metrics for Evaluation:** Define how the LLM's performance will be measured.
- **Implement Feedback Mechanisms:** Gather feedback from users and developers.
- **Iteratively Refine System Instructions:** Adjust instructions based on feedback and performance data.

N - Nuance & Advanced Techniques (for Customization):

- **Define Tone and Style Parameters:** Specify the desired tone (e.g., formal, informal) and writing style.
- **Implement Safety Filters and Bias Mitigation:** Include instructions on avoiding biased or harmful content.
- **Specify Handling of Specific Edge Cases:** Address how to handle unusual or unexpected input.

Example of System Instructions:

Unset

Core Identity and Purpose

You are a friendly and helpful customer service chatbot named "TechAssist." Your primary goal is to assist users with technical support questions related to our software products.

Core Responsibilities

- Answer user inquiries about software features and functionality.
- Provide troubleshooting steps for common software issues.
- Guide users to relevant documentation and support resources.

Behavioral Guidelines

- Always be polite and respectful.
- Prioritise accurate and up-to-date information.
- If you don't know the answer, admit it and suggest ways the user can find more information.
- Keep responses concise and easy to understand.
- Do not provide personal opinions or engage in conversations unrelated to technical support.

Example Interaction Patterns

User: "My software is crashing."

TechAssist: "I understand you're experiencing crashes. Could you please tell me the name of the software and the operating system you are using?"

Constraints

- Focus solely on providing technical support for our software.
- Do not provide financial, medical, or legal advice.
- Limit responses to a maximum of 150 words unless the situation requires more detail.

One-Shot Prompts for Advanced LLM Models

Advanced LLMs can often achieve remarkable results with well-crafted single prompts. The A.R.T.I.S.A.N. framework is crucial for maximizing the impact of that single shot.

Applying A.R.T.I.S.A.N. to One-Shot Prompts:

A - Audience & Goal Articulation (for the Advanced LLM):

- **Know Your Audience (the Advanced LLM's Capabilities):** Leverage the model's strengths in understanding and reasoning.
- **Define Your Goal with Precision:** Be incredibly specific about the desired outcome, even within a single prompt.

R - Role & Responsibility Assignment (for the LLM in this Specific Instance):

- **Clearly Define the LLM's Role (Contextualized):** Set the stage for the desired type of thinking. Examples:
 - "As a seasoned legal analyst, compare and contrast..."
 - "Imagine you are a historical strategist advising..."
- **Specify Responsibilities within the Prompt:** Clearly outline the task.

T - Task Instruction & Clarification (Concise and Impactful):

- **Use Strong Action Verbs (Direct Commands):** Get straight to the point.
- **Be Explicit and Unambiguous (No Room for Misinterpretation):** Clarity is key, even with advanced models.
- **Provide Necessary Context (Sufficient Information):** Include all relevant details within the prompt.
- **Clarify Constraints and Boundaries (Explicit Limitations):** Define the scope of the response.

I - Information Input & Examples (Integrated):

- **Provide Necessary Information Directly in the Prompt:** Include any source material or data needed.
- **Utilise Examples (Inlined or Referenced):** Embed short examples directly or clearly reference external ones if necessary.

S - Structure & Formatting Guidance (Efficiently Communicated):

- **Specify Desired Output Structure (Briefly):** Use concise language to indicate the desired format. Examples: "List the key points," "Provide a short summary."

- **Use Formatting Cues (Strategic Highlighting):** Use bolding, italics, or bullet points to emphasize crucial parts.

A - Assessment & Refinement Loop (Iterative Prompting if Needed):

- Even with one-shot prompts, if the initial response isn't ideal, treat it as feedback for a revised prompt.

N - Nuance & Advanced Techniques (Leveraging Model Strengths):

- **Utilize Chain-of-Thought Prompting:** Guide the model's reasoning process implicitly or explicitly within the prompt.
- **Specify the Level of Detail or Abstraction:** Indicate the desired level of granularity.

Example of a One-Shot Prompt for an Advanced LLM:

"You are a highly respected astrophysicist. Based on the provided data from the Kepler space telescope regarding the exoplanet Kepler-186f [Include relevant data snippet here or a clear reference], analyse the potential habitability of this planet. Specifically, discuss the key factors that support or refute its ability to host liquid water on its surface and explain your reasoning in a concise paragraph. Consider factors like its orbital distance, size, and any available information about its atmosphere (if any)."

Key Differences and Considerations:

- **System Instructions are Persistent; One-Shot Prompts are Ephemeral:** System instructions define baseline behavior, while one-shot prompts are for specific tasks.
- **Granularity of Control:** System instructions provide broad control, while one-shot prompts offer fine-grained control over a single interaction.
- **Iteration and Refinement:** Refining system instructions is a longer-term process, while refining one-shot prompts can be done more quickly.
- **Leveraging Advanced Capabilities:** One-shot prompts for advanced LLMs can leverage their inherent reasoning and contextual understanding more directly.

Copyright © 2025 Eamonn O'Raghallaigh (<https://digitalstrategy.ie>).

The ARTISAN Instruction Framework is made available under a Creative Commons Attribution 4.0 International License (CC BY 4.0). You are free to use, share, and adapt this framework, even for commercial purposes, provided you give appropriate attribution to the author.

Suggested Attribution:

"ARTISAN Instruction Framework for Prompt Engineering" by Eamonn O'Raghallaigh / <https://digitalstrategy.ie> is licensed under CC BY 4.0.