In-Context Learning

How to optimize the results from LLMs for Your Needs

Date: 07/01/25

Author: Ihnaee Choi
AI Specialist, Advisory Solution Consultant

LLM Training And Tuning Is Expensive. Any Alternatives?

From traditional way to fine-tune the model to In-Context Learning

Fine Tuning (Traditional way)

Fine-tune the parameters of the pre-trained model for a specific downstream task using a large (thousands to hundreds of thousands) corpus of labeled data.

Keep training the model via repeated gradient updates.

- Strong performance on many benchmarks.
- Need a new large dataset for each task.
- Potential for poor out-of-distribution generalization
- Potential to explore spurious features of the data

In-Context Learning (LLMs)

Learning of a few examples in the context in generating the answers by LLMs

No training or optimization of the model parameters in the “adaptation step”

Simply give the model a task description as well as none/one/few examples as the input at inference time.

- Only the task description: ZERO-SHOT
- TD + one examples : ONE-SHOT
- TD + a few examples: FEW-SHOT
- No gradient updates are performed.

How Can We Optimize LLMs Without Training And Tuning?

Optimizing LLMs Through Prompting and Tuning

Pre-trained Model (Base LLM Model)
Trained using text data set to predict the next word in a sequence

Prompt Engineering

Zero-Shot Learning

- The prompt provides only a task description or instruction without any input-output examples
- Relying on the model's pre-trained knowledge to generate a response

Few-Shot Learning

- The Prompt includes a small number of input-output examples (~10) to demonstrate the task, allowing the model to generate and perform similar tasks on new inputs

Fine-Tuning

- Fine-tuning requires a substantial collection of human-annotated input-output pairs, where each output is rated or verified for correctness and quality. These examples serve as supervised to adjust the model's parameters for improved task-specific performance.

Reinforcement Learning With Human Feedback

- Human annotators rank multiple model outputs by preference. These rankings train a reward model that scores outputs based on alignment without human intent.
- The base model is fine-tuned using reinforcement learning, optimizing behavior to maximize reward and produce safer, helpful responses.

Low cost and fast iteration
Leverage prior (Pretrained) knowledge
Task control via input design

VS

High resources cost
Require parameter updates
Persistent learning
Alignment and safety optimization

How In-Context Learning works without Tuning?

The secret is a Bayesian Inference Framework

Input prompt
(Context Examples)

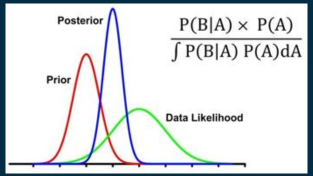
[Input, Output]

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

Latent Concept

θ

Bayesian Inference
(Likelihood)



Prior model probability based on observed data

$$p(\mathrm{output}|\mathrm{prompt}) = \int_{\mathrm{concept}} p(\mathrm{output}|\mathrm{concept}, \mathrm{prompt})\, p(\mathrm{concept}|\mathrm{prompt})\, d(\mathrm{concept})$$

Prior Distribution
(Pretrained model)

Posterior Inference
(Task understanding)

Prediction

Output

Even if ”output” is noisy / randomized, model can infer θ from patterns in input

Four Elements for In-Context Learning?

In-Context Learning Prompt Structure

- A set of example input representing the task type
- e.g., Questions, sentences or commands in the prompt

Input Distribution

- Corresponding outputs that match the input distribution
- e.g., Answers, translations, summaries etc.

Output Distribution

Circulation revenue has increased by 5% in Finland.	\n	Positive
Panostaja did not disclose the purchase price.	\n	Neutral
Paying off the national debt will be extremely painful.	\n	Negative

Format

- The shared structural pattern across all examples
- e.g., "input -> output" format

Input-Output Mapping

- Provide the functional relationship between input and outputs
- e.g., Defines the task implicitly (translation, tag etc.)

**Referred to Stanford*

Example Elements Are Central To In-Context Learning

Best Practices for prompt engineering structure

Must be included	Highly recommended Likely to improve performance significantly			Good to have	
Task	**Context**	**Persona**	**Example**	**Format**	**Tone**
Assign one task ▪ The prompt must include a clear and concise command or instruction, expressed as a verb phrase ▪ To get quality result, only one task should be assigned for one prompt	**Nail relevant Background information** ▪ Specifics of the task ▪ What is the context of the situation? ▪ What is the purpose and objective? ▪ What are the potential risks or concerns? ▪ What are the limitations and guidelines?	**Define role and try role-playing** ▪ As a person, role or job title best suited to solve the problem ▪ The more specific the expert designation, the more industry-specific terminology you can incorporate, and the more precisely you can define the field, the more tailored the response will be.	**Give reference and cite textbook examples** ▪ Feed the referential data ▪ If necessary, multiple sources will be file ▪ URLs, PDFs, Excel, Docx ▪ (e.g., COTs, Few Shot)	**Specify the desired format** ▪ Clearly define the desired format and length of the output ▪ Specify where you need a table, a Markdown file, or provide a detailed content outline	**Set the tone** ▪ Use adjectives to provide example to help the model mimic a specific style

In-Context Learning Methods

Chain of thought (COT)

- Standard prompting techniques (also known as general input-output prompting) do not perform well on complex reasoning tasks (arithmetic reasoning, commonsense reasoning, and symbolic reasoning)
- COT boosts performance of LLMs such non-trivial cases involving reasoning
- CoT incorporates intermediate reasoning steps that can lead to the final output into the prompts. As can be seen from the example below.

Self-consistency COT

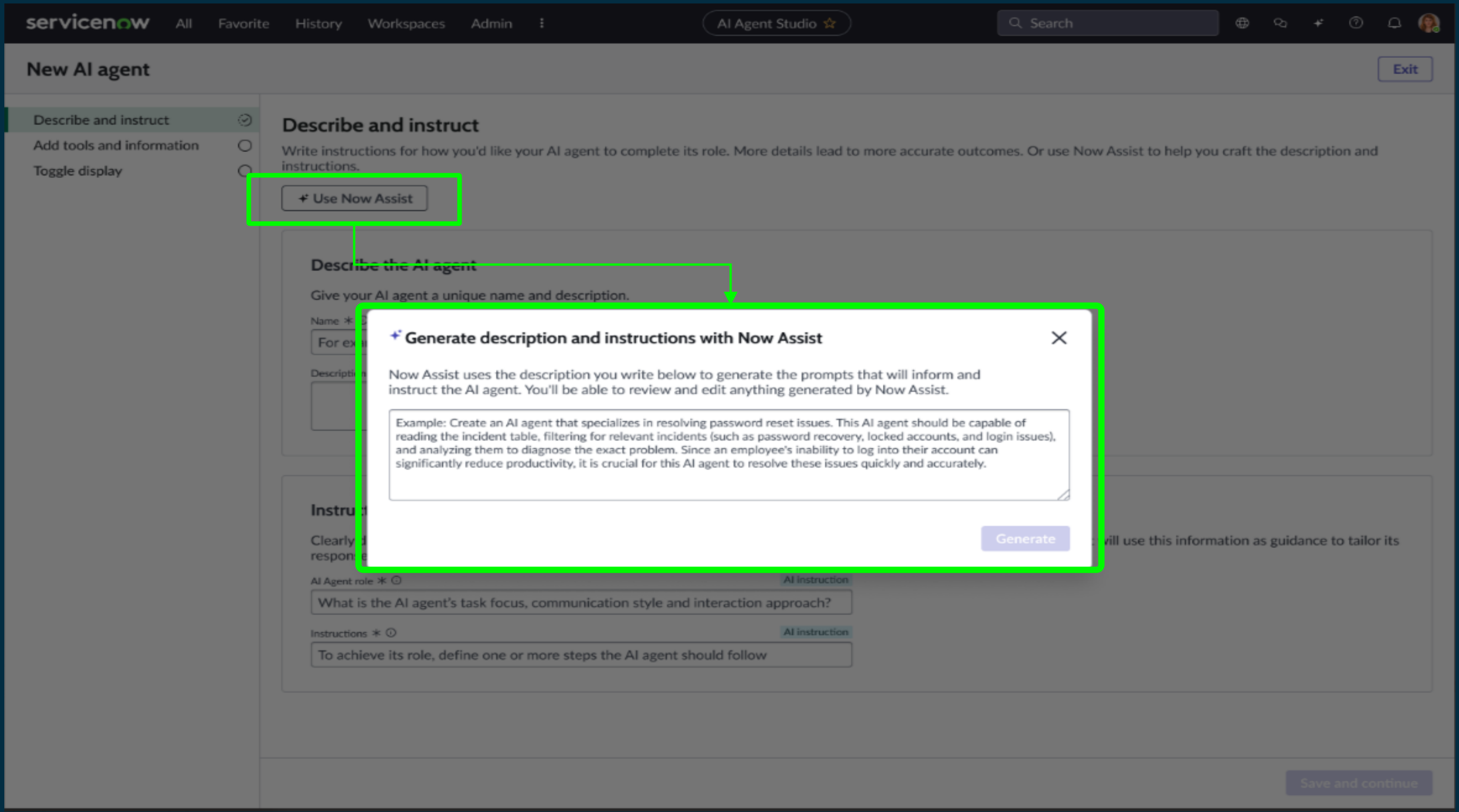
- It works by first generating multiple chains of thought, each of which is a sequence of steps that leads to a possible answer to the question. The LLM then selects the most consistent answer by taking the majority vote from the generated chains of thought.

Tree of Thoughts

- ToT works by decomposing a problem into a tree of thoughts, where each thought is a self-contained unit of reasoning. The LLM is then prompted to generate a sequence of thoughts that leads to a solution to the problem.
- The ToT framework has been shown to be effective in improving the performance of LLMs on a variety of tasks, including math reasoning, creative writing, and game playing. In one study, ToT was shown to boost the performance of an LLM on the Game of 24 by 74%.

Too Complex? Now Assist Delivers a One-Click Solution

*** From Yokohama*



- **Now Assist** provides a one-click solution for rewriting prompts based on a user's minimal input.
- In AI Agent Studio, when a user creates a user case with a draft prompt(instructions), Now Assist automatically populates the relevant fields and suggests a refined prompt as placeholder text.