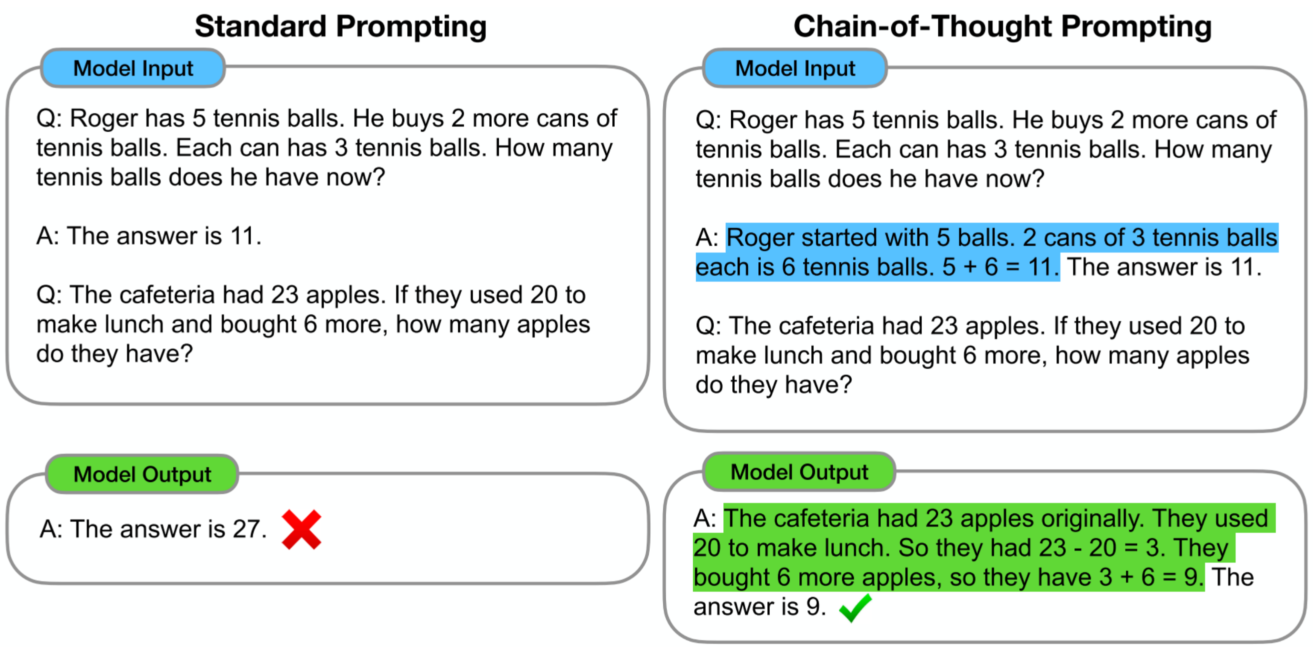
GenAI for Software Engineering 2025

Reasoning and Acting

Jorge Melegati

Previously in this course…

- Chain-of-thought
  - Suggest intermediate natural language steps leading to the final output
- Limitation:
  - Limited to its internal representations learned during training



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted

Source: Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models" (2022)

How do humans do?

- Humans combine task-oriented actions with verbal reasoning (inner speech)

- Combination of reasoning and acting

Example: preparing bread

- Steps:
  - Choose a recipe
  - Combine the ingredients
  - Check if more water is needed
  - Knead the dough
  - First rise
  - Shape the dough
  - Second rise
  - Bake it!

Idea: give similar possibilities to an LLM model

Published as a conference paper at ICLR 2023

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao*,1, Jeffrey Zhao2, Dian Yu2, Nan Du2, Izhak Shafran2, Karthik Narasimhan1, Yuan Cao2

1Department of Computer Science, Princeton University
2Google Research, Brain team
1{shunyuy,karthikn}@princeton.edu
2{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

ABSTRACT

While large language models (LLMs) have demonstrated impressive performance across tasks in language understanding and interactive decision making, their abilities for reasoning (e.g. chain-of-thought prompting) and acting (e.g. action plan generation) have primarily been studied as separate topics. In this paper, we explore the use of LLMs to generate both reasoning traces and task-specific actions in an interleaved manner, allowing for greater synergy between the two: reasoning traces help the model induce, track, and update action plans as well as handle exceptions, while actions allow it to interface with and gather additional information from external sources such as knowledge bases or environments. We apply our approach, named ReAct, to a diverse set of language and decision making tasks and demonstrate its effectiveness over state-of-the-art baselines in addition to improved human interpretability and trustworthiness. Concretely, on question answering (HotpotQA) and fact verification (Fever), ReAct overcomes prevalent issues of hallucination and error propagation in chain-of-thought reasoning by interacting with a simple Wikipedia API, and generating human-like task-solving trajectories that are more interpretable than baselines without reasoning traces. Furthermore, on two interactive decision making benchmarks (ALFWorld and WebShop), ReAct outperforms imitation and reinforcement learning methods by an absolute success rate of 34% and 10% respectively, while being prompted with only one or two in-context examples.

1 INTRODUCTION

ReAct

- “a general paradigm to combine reasoning and acting with language models for solving diverse language reasoning and decision making tasks”
- “Prompts models to generate verbal reasoning traces and decision-making tasks”
- Including interacting with the external environment

Function/tool calling

- Ability to retrieve information from external tools, e.g., APIs or performing actions

Example with OpenAI api

- “What is the weather like in Paris today?”
- https://platform.openai.com/docs/guides/function-calling

RAG vs Function calling

RAG

- Use: consider relevant information from large dataset
- Information is retrieved before interacting with the LLM (vector databases)
- Information is added in the original prompt

Function calling

- Use: obtain information from external sources
- Information is retrieved based on the request of the model
- Information is added in a following prompt. The original prompt should contain the definitions of the functions

Example: Getting information about a Git repository

- Problem: summarize the history of changes of a Git

- Solution: use a function call to provide the information need about the changes to be used in the summary

- Open in Google Colab the file FunctionCallingForGit.ipynb from the repository melegati/gen4ai-course
  - If you prefer, you can also run locally!

Multiple ReAct agents

- Combine multiple agents for specific subtasks
- Maybe with different capabilities: CoT, RAG, function calling, etc.
  - Even without any AI capability
- Specific libraries for implementation
  - LangGraph

Summary of techniques seen

- Standard prompt engineering

Always!

- Chain-of-thought

Almost always!

- RAG
- Function calling

Exercise: Let’s recall the prioritization problem!

Roles:

- Product Owner (PO)
- Quality Assurance specialist (QA)
- Developers (DEV)

How can we improve this using RAG and function calling?

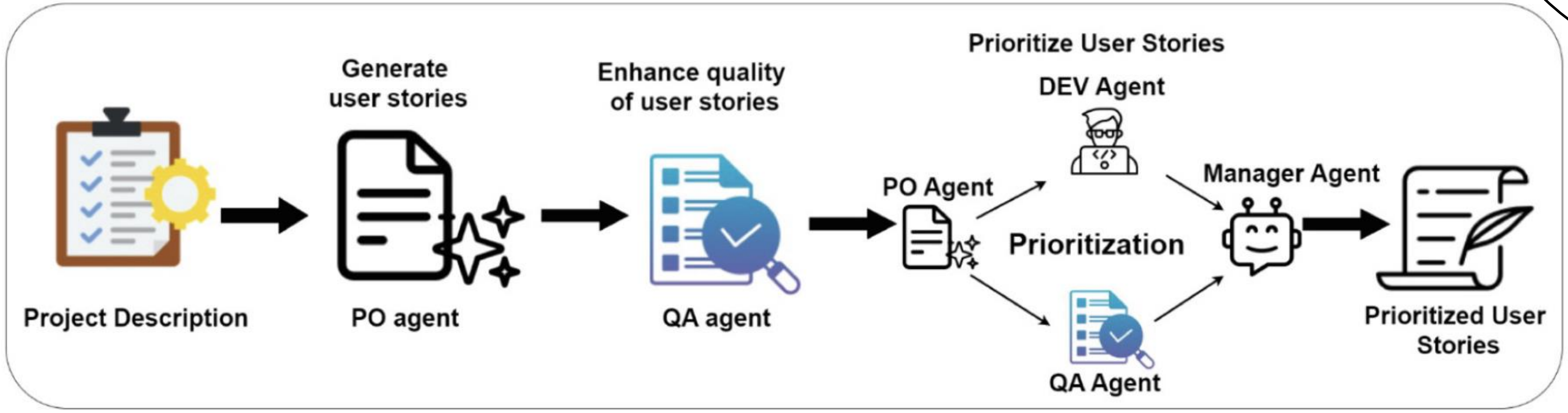


Fig. 1. Process Model of LLM Agents for User Story Generation, Quality Check, and Prioritization.