Automated Graph of Thoughts: Orchestrating Elaborate Problem Solving with LLMs

Degree programme : BSc in Computer Science
Thesis advisor : Peter Alfred von Niederhäusern
Expert : Patrick Joset

This work examines the extent to which autonomous (RL) agents can generate and traverse Graphs of Thoughts (GoT) to achieve optimal results in selected tasks.

Manuel Noah Riesen
Data Engineering
manuel.riesen0@gmail.com

Introduction

Graph of Thoughts (GoT) is a generalized form of recent prompting paradigms for Large Language Models (LLMs), where thoughts are structured as an arbitrary graph. By executing a graph of operations, the GoT is populated with thoughts until the desired result is achieved.
This work examines the extent to which Reinforcement Learning (RL) agents can generate and traverse graphs of thoughts to solve a given task.
While the operations are formulated by humans, the agent composes the operations and prompts the LLM.

The task **sum list** acts as the primary example in this work, as it captures the overall problem to solve with GoT. The language model‘s result is only reliable to a certain number of single-digit integers in the list. To reliably solve the task, a divide-and-conquer strategy must be applied (**split**, **sum** and **merge**).

Methodology

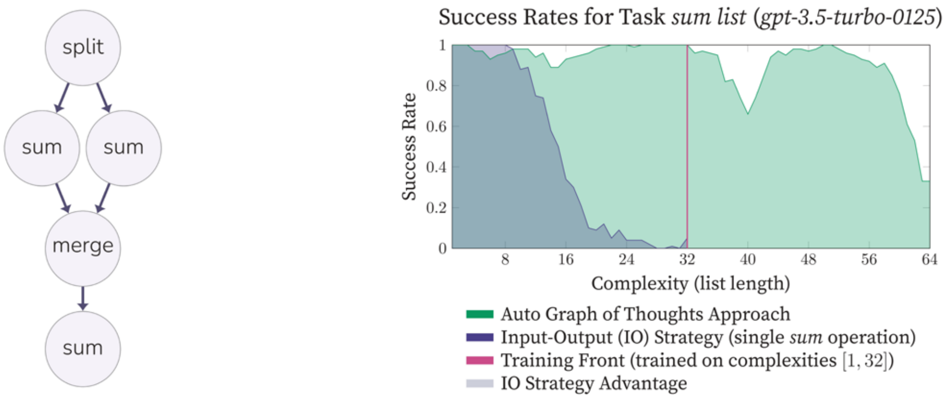
The work is driven by the conduction of experiments. Initially, simple baseline algorithms (Random, Simulated Annealing) are applied. A custom RL environment is implemented and tested by using a simple RL algorithm. Then, more complex RL algorithms are leveraged to solve the example task. In a first experiment, it is attempted to train an agent that solves the example task with a fixed complexity. A second experiment is conducted to examine how the agent performs when facing a task with variable complexity (both seen and unseen complexities).

Results

The results show that, under certain constraints, it is possible for an agent to solve the example task, even with a variable complexity. The reliability is dependent on the complexity. As a side-product, a custom GoT framework is implemented. It provides a typed API to define tasks and operations (DSL) and allows the manual construction of graphs of operations.

Discussion

Potentially, the automated GoT approach is applicable to various tasks that can be solved by language models, such as merging documents, counting keywords in a text or summarizing large documents.



Example graph of operations (left) and success rates (right) for the task «sum list»