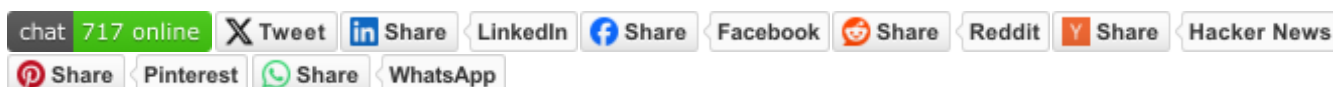




Tree of Thoughts



Paper link [Author's implementation](#)

Introduction

Tree of Thoughts (ToT) is a powerful and flexible algorithm that significantly advances model reasoning by up to 70%. This plug-and-play version allows you to connect your own models and experience superintelligence!

Install

```
1 pip install tree-of-thoughts
```

Usage

```
1 import os
2 from tree_of_thoughts import ToTAgent, MonteCarloSearch
3 from dotenv import load_dotenv
4 from swarms import Agent, OpenAIChat
5
6 load_dotenv()
7
8 # Get the API key from the environment
9 api_key = os.environ.get("OPENAI_API_KEY")
```

```

10
11 # Initialize an agent from swarms
12 agent = Agent(
13     agent_name="tree_of_thoughts",
14     agent_description="This agent uses the tree_of_thoughts library to
15         generate thoughts.",
16     system_prompt=None,
17     llm = OpenAIChat(),
18 )
19 # Initialize the ToTAgent class with the API key
20 model = ToTAgent(
21     agent,
22     strategy="cot",
23     evaluation_strategy="value",
24     enable_react=True,
25     k=3,
26 )
27
28
29 # Initialize the MonteCarloSearch class with the model
30 tree_of_thoughts = MonteCarloSearch(model)
31
32 # Define the initial prompt
33 initial_prompt = """
34
35
36 Input: 2 8 8 14
37 Possible next steps:
38 2 + 8 = 10 (left: 8 10 14)
39 8 / 2 = 4 (left: 4 8 14)
40 14 + 2 = 16 (left: 8 8 16)
41 2 * 8 = 16 (left: 8 14 16)
42 8 - 2 = 6 (left: 6 8 14)
43 14 - 8 = 6 (left: 2 6 8)
44 14 / 2 = 7 (left: 7 8 8)
45 14 - 2 = 12 (left: 8 8 12)
46 Input: use 4 numbers and basic arithmetic operations (+-*/ ) to obtain
47     24 in 1 equation
48 Possible next steps:
49 """
50 # Define the number of thoughts to generate
51 num_thoughts = 1
52 max_steps = 3
53 max_states = 4
54 pruning_threshold = 0.5
55
56
57 # Generate the thoughts
58 solution = tree_of_thoughts.solve(

```

```
59     initial_prompt=initial_prompt,
60     num_thoughts=num_thoughts,
61     max_steps=max_steps,
62     max_states=max_states,
63     pruning_threshold=pruning_threshold,
64     # sleep_time=sleep_time
65 )
66
67 print(f"Solution: {solution}")
```

ToT with HF LLM

To run Hugging Face Transformers with Tree of Thoughts:

```
1  import os
2  from tree_of_thoughts import ToTAgent, MonteCarloSearch
3  from dotenv import load_dotenv
4  from swarms import Agent, HuggingfaceLLM
5
6  load_dotenv()
7
8  # Get the API key from the environment
9  api_key = os.environ.get("OPENAI_API_KEY")
10
11 # Initialize an agent from swarms
12 agent = Agent(
13     agent_name="tree_of_thoughts",
14     agent_description=(
15         "This agent uses the tree_of_thoughts library to generate
16         thoughts."
17     ),
18     system_prompt=None,
19     llm=HuggingfaceLLM(
20         "EleutherAI/gpt-neo-2.7B",
21     ),
22 )
23
24 # Initialize the ToTAgent class with the API key
25 model = ToTAgent(
26     agent,
27     strategy="cot",
28     evaluation_strategy="value",
29     enable_react=True,
30     k=3,
31 )
32
33 # Initialize the MonteCarloSearch class with the model
34 tree_of_thoughts = MonteCarloSearch(model)
```

```

35
36 # Define the initial prompt
37 initial_prompt = ""
38
39
40 Input: 2 8 8 14
41 Possible next steps:
42 2 + 8 = 10 (left: 8 10 14)
43 8 / 2 = 4 (left: 4 8 14)
44 14 + 2 = 16 (left: 8 8 16)
45 2 * 8 = 16 (left: 8 14 16)
46 8 - 2 = 6 (left: 6 8 14)
47 14 - 8 = 6 (left: 2 6 8)
48 14 / 2 = 7 (left: 7 8 8)
49 14 - 2 = 12 (left: 8 8 12)
50 Input: use 4 numbers and basic arithmetic operations (+-*/) to obtain
    24 in 1 equation
51 Possible next steps:
52 ""
53
54 # Define the number of thoughts to generate
55 num_thoughts = 1
56 max_steps = 3
57 max_states = 4
58 pruning_threshold = 0.5
59
60
61 # Generate the thoughts
62 solution = tree_of_thoughts.solve(
63     initial_prompt=initial_prompt,
64     num_thoughts=num_thoughts,
65     max_steps=max_steps,
66     max_states=max_states,
67     pruning_threshold=pruning_threshold,
68     # sleep_time=sleep_time
69 )
70
71 print(f"Solution: {solution}")

```

Basic Prompts

- Copy and paste this into your llm!

```

1 "Three experts with exceptional logical thinking skills are
   collaboratively answering a question using the tree of thoughts
   method. Each expert will share their thought process in detail,
   taking into account the previous thoughts of others and admitting
   any errors. They will iteratively refine and expand upon each other's
   ideas, giving credit where it's due. The process continues until a

```

conclusive answer is found. Organize the entire response in a markdown table format. The task is:

Acknowledgements

Thanks to: Shunyu Yao Princeton University, Dian Yu Google DeepMind, Jeffrey Zhao, Google DeepMind, Izhak Shafran Google DeepMind, Thomas L. Griffiths, Princeton University, Yuan Cao Google DeepMind, Karthik Narasimha, Princeton University for sharing this amazing work with the world!

And, thanks to Phil Wang or Lucidrains for inspiring me to devote myself to open source AI Research

License

Apache