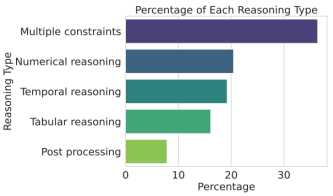
Sentient Challenge – Siuuupremacy

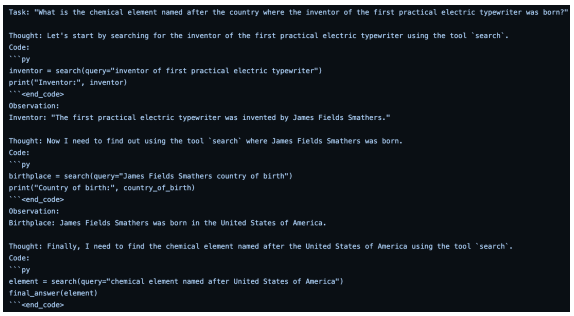
Our exploratory analysis on how the base model performs on the FRAMES dataset shows that the base model performs poorly on 2 types of questions that are mostly responsible for the poor performance of OpenDeepSearch (multiple constraints and numerical reasoning). We therefore decided to focus on improving model performance on multiple constraint questions.

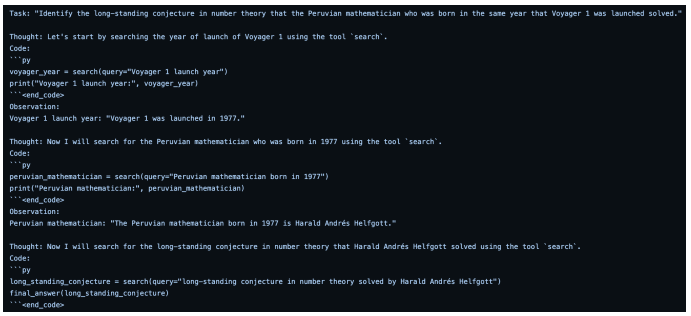


After singling out the model components we could improve upon, we concluded that effective strategies would entail either changing the prompts to alter the behavior of the agents and the OpenDeepSearch module or changing the actual tool architecture. We modify the agent behavior in two ways: DeCRIM pipeline for self-correction and refined instruction for query breakdown.

The first approach is to integrate a DeCRIM pipeline consisting of 3 main components (Decompose, Critique and Refine), which according to the original paper could potentially improve results by about 8%. The main idea is that we ask the model to define the constraints of the questions at the beginning. Upon receiving all the context from the search tool, the agent attempts to find the answer within the context and check whether the constraints are met or not. If it succeeds the final answer is returned; otherwise, the reasoning process is restarted by feeding the model the original task together with the feedback received from the failed attempt.

The second approach to improving the model's ability to handle complex queries stems from observing that the base model often makes errors due to incorrect query division. For instance, in the first entry of the FRAMES dataset, the model incorrectly ranks buildings by height (e.g., top 3, 5, or 10) instead of identifying the absolute position of the building in question compared to the tallest buildings. To address this, we focused on training the model with more intricate queries, similar to those in the FRAMES dataset, to enhance its accuracy. Our intuition is that FRAMES tasks involve multi-hop reasoning, which is significantly more complex than the few-shot prompts outlined in the guidelines. As a result, the model may struggle with reasoning, leading to hallucinations or overlooked details. To mitigate this, we designed two challenging tasks in the CodeAgent prompt that align with the complexity and style of the FRAMES dataset.





Additionally, we experienced API key outage for crucial processes like the Jina reranker. To combat this, we implemented our own embedder using fireworks API which we had “unlimited” use of, notably the `nomic-ai/nomic-embed-text-v1.5`

Results

Due to API usage limits, we have yet to consolidate our two main approaches, despite exploratory analysis suggesting promising results. The best result we achieved uses enhanced query breakdown with CodeAgent and custom reranker, with the evaluation score of 0.6529.