Recursive Self-Improvement

Tero Keski-Valkama

January 27, 2024

Introduction: Tero Keski-Valkama

- Tero Keski-Valkama is an AI practitioner with over 20 years of experience spanning four countries, currently living in Spain.
- Worked with machine vision, complex control, SLAM, sensor fusion, semantic web, perceptrons, genetic algorithms, simulated annealing, SVMs, gradient boosting, deep learning, deep reinforcement learning, GANs, CNNs, Transformers, **LLMs**, **AGI**, embodiment, meta-learning, ...
- Robotics, pre-LLM chatbots, **LLM chatbots**, facial emotion recognition, automatic mapping, logistics, supply chain, ...
- He has authored over 20 patents in the topic among countless other publications.
- Authored the first correct open source Google WaveNet implementation, the first communist AI, cofounded the second largest recurring AI event in Finland (AI Morning), ...

Why Is This a Big Deal?

Chatbots is a misnomer – They don't just chat. They can control other machines, adapt/learn, and most importantly, they integrate together societies of people and machines. They are already as intelligent as humans in many important topics, and very soon surpass humans in every cognitive task.

The **Technological Singularity** has started, and along it the societies are now waking up to a sudden and extreme appetite for data, knowledge, network connectivity and chips.

> *"The fabric tunnel that stretched out behind it was a* **'camera tunnel...'** *The shredded fragments of books and magazines flew down the tunnel like leaves in a tornado, twisting and tumbling. The inside of the fabric was stiched with thousands of tiny cameras. The shreds were being photographed again and again, from every angle and orientation, till finally the torn leaves dropped into a bin just in front of Robert."*
>
> *- Vernor Vinge, Rainbows End*

Tips and Tricks

You would typically integrate a chatbot to perform a task by making a small **Python** code which calls the **chatbot API** when something happens, give it some amount of **carefully designed context**, and utilize its responses to **make decisions** or **generate answers**.

To limit the amount of context you give to the chatbot for every task you make it to do you can **decompose** the knowledge **hierarchically**. In effect you ask the chatbot whether the context it got relates to some topic, and depending on its answer, you clear the session and start from scratch, now performing a **subtask**. You do this for all separate subtasks and then collect together the answers in some way.

Alternatively you can make your **knowledge base** searchable by the chatbot so that it can ask specific information it needs from it only when it needs it.