



## Assignment 4A: Generative Agents with LLMs (mandatory, 5p)

### An LLM Agent that can Read Papers

**Background:** AI assistants are agents that can recommend actions to users and trigger actions autonomously on their behalf. LLMs have revolutionized the way interactive assistants are built. LLMs have made it significantly easier to develop an assistant. Assistants can now make inferences from textually presented materials without insisting on hand-crafted knowledge like rules or graphs. Some prompting and architecting is needed, obviously. The new challenge is how to guide the assistant to draw these inferences in the right way. In this assignment, we look at the creation of an assistant using a breakthrough framework called *Generative Agents*.

**Task:** The theme of this week's assignment is research. Since you have built the Reading assistant in the class, your goal is to create a *Reading Agent* that can interact with the reading assistant. In particular, your agent:

1. Has a goal of understanding a research subject by reading a paper.
2. Is given a sequence of words as main input (e.g., each word from a PDF).
3. Should generate a reading behavior (e.g., similar to the 'new\_observations.txt' used for testing the reading assistant) and update its understanding of the research subject.
4. Occasionally receives suggestion from the reading assistant and should reason with its responsive action (e.g., ignore and continue reading or respond).
5. Should also reason with when to stop reading (e.g., fully understood the research subject).

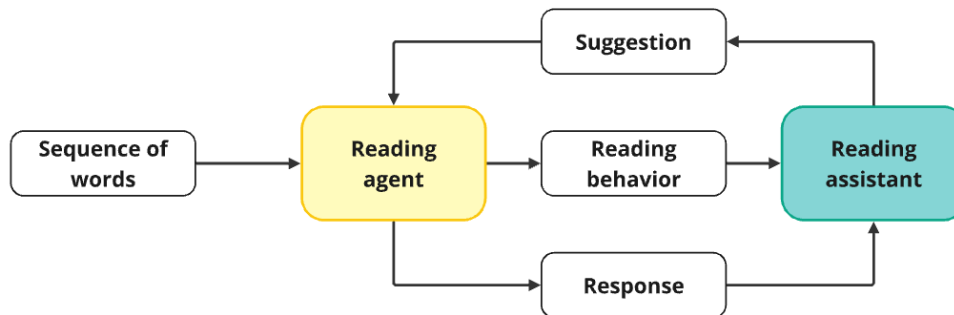


Figure 1. Your task is implementing the Reading Agent (the yellow box).

**Approach:** You need to create a Python (or JavaScript) program that accesses OpenAI’s GPT via its API (further tutorials below). Note: direct prompt engineering will not suffice, as a coherent user profile should be formed in order to make meaningful inferences.

**Preparations:** First, choose how your reading agent should reason to decide whether it has acquired “enough” understanding of the reading subject or not. This is a tricky problem, and we recommend defining the quality of understanding as precisely as possible (e.g., What should the agent understand by reading the title of a paper? What is enough understanding to choose a quality research topic? How should agent judge that?). Describe these goals and the scope of your agent in your report. Then, if GPTs are not familiar to you, familiarize with them:

- Intro to LLMs by Karpathy: [https://www.youtube.com/watch?v=zjkBMFhNj\\_g](https://www.youtube.com/watch?v=zjkBMFhNj_g)
- GPT intro: <https://www.youtube.com/watch?v=MQnJZuBGmSQ> (14 mins)

Then, create an account for Open AI, unless you have one. Find your API key. Note that Open.ai has a cap to the number of tokens you can use for free.

**Getting started:** You can use the Python skeleton with API calls that we offer in the JupyterLab notebook. More help:

- OpenAI API Docs: <https://platform.openai.com/docs/concepts>
- OpenAI API Quick start tutorial: <https://platform.openai.com/docs/quickstart>
- <https://jman4190.medium.com/how-to-build-a-gpt-3-chatbot-with-python-7b83e55805e6>

**Reporting results:** In addition to a report (format as usual), we ask you to record a video (screen capture) in your submission archive. The video should include at least 4 interactions between the reading agent and the reading assistant using *representative* inputs. Pay attention to how you choose the inputs: on the one hand, you should avoid cherry picking; on the other, do not show failure cases only. Pick inputs that match your stated use case. Audio narration is not needed. In your PDF report, also include a brief assessment of your solution.

**Grading:**

- Convincing and well-scoped use case presented [+0.5 p]
- A reasonable approach outlined and justified [+1 p]

- Implementation [+2 p]
- Results are meaningful [+1 p]
- Results are critically discussed [+0.5 p]
- Video missing [-3 p]
- We may provide bonus points [added on top of the 5p maximum] for particularly well-thought and executed solutions.