Accelerated Generative AI Learning Path for GENIE

The purpose of this learning path is to introduce you to basic generative AI concepts that you will need for working in the GENIE research group. This is an accelerated learning path, meaning it takes you through the core ideas in 1 week.

After completing this learning path, you will be able to:

* Understand the difference between generative and predictive AI
* Describe what generative AI and LLMs are capable of
* Understand the building blocks and workflow of a typical generative AI application
* Develop a basic LLM-based application using prompt engineering, retrieval augmented generation (RAG), and basic evaluation tools.

**Day 1: Basics of Generative AI and your first LLM application**

* Watch all the videos in Week 1 of the following Coursera course: <https://www.coursera.org/learn/generative-ai-with-llms>
* Read through the introduction section of [this prompt engineering guide.](https://www.promptingguide.ai/introduction)
* Read up on prompting GPT 3.5 over an API [here](https://www.promptingguide.ai/models/chatgpt) and run both the notebooks given at the end of the article (GPT 3.5 prompting with and without Langchain). You will need an OpenAPI key which will be provided to you.

**Day 2: Retrieval Augmented Generation**

* Complete this course from DeepLearning.AI for Retrieval Augmented Generation: [Langchain: Chat with your Data | DeepLearning.AI](http://deeplearning.ai)
* Following the notebook in the course, set up your own RAG pipeline using Langchain and the OpenAI API. Demonstrate a simple Q&A system over any documents ingested into your pipeline.
* Now use the [RAGAS](https://docs.ragas.io/en/stable/index.html) framework to incorporate some common RAG evaluation metrics into your notebook. You can use [this](https://docs.ragas.io/en/stable/howtos/integrations/langchain.html) guide to help you.
* Read up on LLM Evaluation [here](https://llmshowto.com/blog/llm-test-frameworks) and incorporate at least one more evaluation framework in your RAG pipeline.

**Day 3: Fine Tuning and RLHF**

* Complete this course from DeepLearning.AI for understanding the idea of fine-tuning LLMs and the different types of fine-tuning: [Fine-tuning Large Language Models | DeepLearning.AI](http://deeplearning.ai)
* Complete [this](https://learn.deeplearning.ai/courses/reinforcement-learning-from-human-feedback/lesson/1/introduction) course from DeepLearning.AI for understanding the idea of reinforcement learning from human feedback.
* Run the notebooks from both the above courses yourself and try making changes to the datasets, parameters and models to understand the impact of different steps in the tuning process.

**Day 4: Building Blocks of Generative AI Applications**

* Now that you have covered the basic ideas in generative AI, you can understand the generative AI application development lifecycle described [here](https://www.coursera.org/learn/generative-ai-for-everyone/lecture/t2aQM/lifecycle-of-a-generative-ai-project).
* Thoroughly go through [this mindmap](https://www.mindmeister.com/app/map/3171461222?t=muwYOOjdYe) capturing the building blocks of basic-to-advanced generative AI applications - explore the links and understand the main ideas at each step. You will be asked to present your understanding of the space in your meeting with your mentor upon completion of the learning path.

**Day 5: Advanced Features: Tool Calling and Advanced RAG**

* Complete this course on giving LLMs the power to call functions using Langchain and OpenAI’s tool calling: [Functions, Tools and Agents Using Langchain | DeepLearning.a](https://learn.deeplearning.ai/courses/functions-tools-agents-langchain/lesson/1/introduction)i and try out the notebook.
* Read up on advanced RAG [here](https://towardsdatascience.com/advanced-retrieval-augmented-generation-from-theory-to-llamaindex-implementation-4de1464a9930).
* Put together everything you have learned in this week by adding more features to the RAG application you built on Day 2 (more advanced prompt engineering, tool calling and Langchain agents, and some advanced RAG features).