AI Bootcamp

Emerging Topics in AI

Module 22 Day 1

Welcome

Class Objectives

By the end of class, you will be able to:

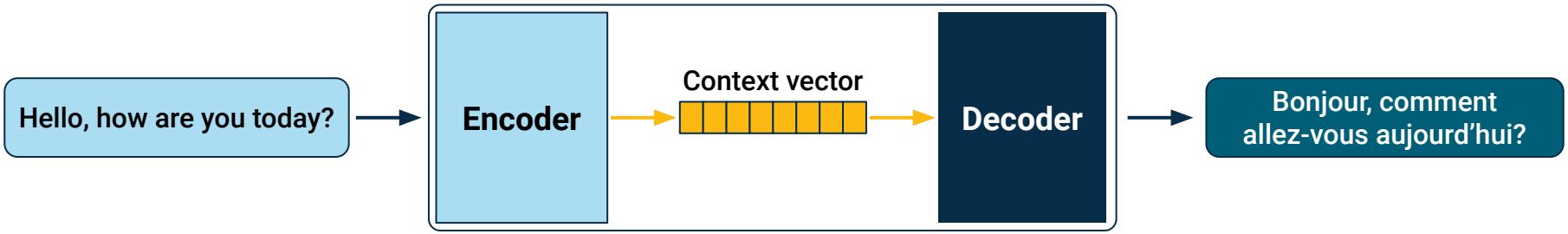
1. Use LangChain classes to build GPT-based applications.
2. Integrate external data from documents and APIs into LangChain programs.
3. Use conversational memory within LangChain applications.

Instructor **Demonstration**

Overview of LangChain and ChatGPT

Transformer models

Encoder-decoder architectures



LangChain is a Python framework that makes it easy to build LLM-powered applications with functions that extend beyond the core LLMs.

LangChain functionalities

LangChain offers six key functionalities:

1. **Models** allow you to access other models from OpenAI, HuggingFace, and more.
2. **Prompts** provide prompt templates for LLMs.
3. **Chains** are sets of calls executed in sequence to combine multiple parts of the program. For example, you could combine an LLM and a prompt template in a chain, but there are endless chain combinations you could use to execute your desired functionality.
4. **Memory** stores the history of a model's outputs for later use.
5. **Indices** are utility functions that allow you to combine the model with your own text data from PDFs, CSV files, and more.
6. **Agents and Tools** set up agents powered by LLMs that can use tools such as Google search and Wikipedia, or connect to applications like Gmail, Jira, and Slack.

LangChain

Chains

01

Chains are individual tools that can be used individually or combined.

02

One chain contains a series of calls to LLMs, tools, or data preprocessing.

03

There are many different chains fit for different purposes already in the LangChain ecosystem, but you can create your own too.

LangChain

Chains

1. Chains are individual tools that can be used individually or combined.
2. One chain contains a series of calls to LLMs, tools, or data preprocessing.
3. There are many different chains fit for different purposes already in the LangChain ecosystem, but you can create your own too.

Some off-the-shelf chains include:

1. `LLMMath` converts a user question to a math problem and then executes it.
2. `LLMSummarizationChecker` can be used to verify the accuracy of another LLM-generated summary.
3. `ConstitutionalChain` restricts the response of an LLM in a question-answer model according to some rule specified by the user.
4. `APIChain` converts a user query to an API request and passes the retrieved response to an LLM for a response.

Instructor **Demonstration**

Introduction to LangChain

Activity:

LangChain Practice

In this activity, you will be extending the recipe suggestion program to allow for diet types and dietary restrictions. You will use **`LLMChain`** and **`ConstitutionalChain`** to ensure that the recipes suggested fit the user's needs.

Suggested Time:
15 Minutes

Time’s up!

Let’s review

Questions?

Break

15 mins

Instructor **Demonstration**

External Sources

Activity:

API Chat

In this activity, you will connect LangChain with the Open Library API to allow a user to ask questions about books and authors.

Suggested Time:
20 Minutes

Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Conversational Memory

Activity:

Event Planner

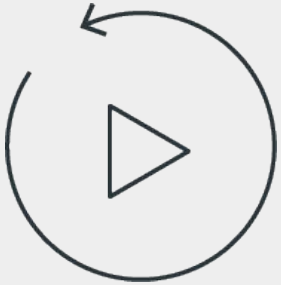
In this activity, you will create an AI event planner that takes a group of people's likes and dislikes into consideration when planning an event for them.

Suggested Time:
20 Minutes

Time’s up!

Let’s review

Questions?



Let’s **recap**

Recap

After today's lesson you are able to:

1. Use LangChain classes to build GPT-based applications.
2. Integrate external data from documents and APIs into LangChain programs.
3. Use conversational memory within LangChain applications.

Next

In the next lesson, you will use LangChain to refine your approach to inputs and outputs in LLM applications.

Questions?

The End