Generative AI Foundations for Engineers

Clair J. Sullivan
clair@clairsullivan.com

CLAIR SULLIVAN
& ASSOCIATES

Introduction

Agenda

- Brief introduction to the people in the room
- Structure for today
  - *Brief* lectures
  - Activities
  - Tasks (what is scored!)
- Module 1: Generative AI Foundations & LangChain Basics
- Module 2: Prompt Engineering & Model Customization
- Module 3: Retrieval-Augmented Generation (RAG) & External Data Integration
- Module 4: Agents & Tool Use
- Wrap up, announcement of winner!

A brief survey:

PollEv.com/clairsullivan399

Schedule

9:00 - 9:15	Introduction
9:15 - 9:30	Module 1: Lecture
9:30 - 10:30	Module 1: Hands On
10:30 - 10:40	Module 2: Lecture
10:40 - 12:00	Module 2: Hands On
12:00 - 1:00	LUNCH
1:00 - 1:10	Module 3: Lecture
1:10 - 2:25	Module 3: Hands On
2:25 - 2:35	Module 4: Lecture
2:35 - 3:50	Module 4: Hands On
3:50 - 5:00	Wrap up, awards!

Module 1: Generative AI Foundations & LangChain Basics

What is Generative AI (GenAI)?

- Traditional AI (AKA machine learning) focuses on classification, prediction
- GenAI **creates** new content
- Trained on large datasets using deep learning to learn patterns and generate novel output
- Example of GenAI in action?

How Large Language Models (LLMs) Work

- Transformer models: the backbone of LLMs
  - Captures relationships between words across long texts
  - Key innovation in models like GPT, BERT, etc.
- Text is broken into **tokens**
- LLMs have a maximum number of tokens they can process: **context window**
  - When exceeded, older tokens are forgotten (“sliding window effect”)
  - Varies by model
  - Longer context ≠ perfect memory
- LLMs are asked to do things through their **prompts**
- Limitations of LLMs
  - **Hallucinations**
  - Training data limitations (date, subject matter)
  - Computation cost

Introduction to LangChain

- Framework (Python, JS) for building applications powered by LLMs
- Why use it?
  - Open source
  - Simplifies development
  - Has wrappers for most common LLMs, cloud platforms
  - Manages context
  - Integrates with many data types, APIs
  - Chains, agents, tools easily created
- Cons of using LangChain
  - Debugging issues harder compared to direct API calls
  - Latency
  - In *active* state of development

Introduction to AWS Bedrock

- Fully managed service that provides foundation models via APIs
  - Anthropic, Meta, Mistral, Amazon Titan
- Enables creation of GenAI applications without managed infrastructure
- Seamless AWS integration with tools like S3, SageMaker, IAM, etc.

Structure for the Hands On Work

- Three topic areas
  - Code generation
  - Structured data analysis
  - Unstructured data analysis
- You don’t have to stick with one topic area or submit solutions for every question
- Start by going through the “Activity” notebooks
- You will be scored based on the “Tasks”

Ground Rules

- **Have fun!!!**
- Experiment
- Engage with everyone on your team
- Extra points can be awarded at any time for great *shared* learnings
- Extra points awarded for early submissions

Some Notes for Module 1

- Don’t worry (too much) about rewriting the prompts
  - Just try to ask good questions through the Human Prompt
- Some problems might be easy to solve with simple coding, but let’s stick to making the LLMs do it for us
- All code, slides can be found on the repo

https://github.com/ClairSullivan-Associates/genai_foundations

How to get started with AWS Workshop