LLMs, LangChain and Conversational Flows

22 Maggio 2025

Michele Ferro
michele.ferro@aitho.it

Data Scientist & BE Developer

Agenda

- **From ML to NLP and GenAI**
  The rise of a new AI era
- **LangChain**
  Managing conversational flows with LLMs
- **LangGraph**
  Building stateful and dynamic conversational flows
- **Example Notebook**

From ML to NLP and GenAI

The rise of a new AI era

aitho

From classic ML to language understanding

- **Classic Machine Learning**
  - Works well with structured data (tables, numbers, labels)
  - Uses algorithms like regression, decision trees, SVMs
  - Requires **manual feature engineering** and **domain expertise**
- **The Challenge**
  - Language is **unstructured**, **ambiguous** and **context-dependent**
  - Traditional ML struggles to capture meaning, syntax and semantics

NLP Evolution, GenAI and the rise of LLMs

Natural Language Processing (NLP)	Generative AI (GenAI)	Large Language Models (LLMs)
– Field of AI that deals with the automatic **processing of natural language** (text or human voice) – **Early approaches:** rule-based systems, bag-of-words, TF-IDF – **Then:** statistical models, followed by word embeddings (Word2Vec, Glove) – **Breakthrough: *Transformers* (2017) →** enabled true deep contextual understanding	– Field of AI that focuses on **generating content** (i.e. text, images, code, audio, video) – Produces **original**, high-quality outputs – Enables **multimodal interaction** (e.g. text-to-image, code from natural language) – Fuels next-gen applications in **productivity**, **creativity** and **automation**	– Foundation models trained on **massive text corpora** – Can **generate**, **translate**, **summarize** and **reason** over language – Examples: ***GPT*** (OpenAI), Gemini (Google), ***LLaMA*** (Meta), ***Claude*** (Anthropic), ***Mistral*** (Mistral AI)

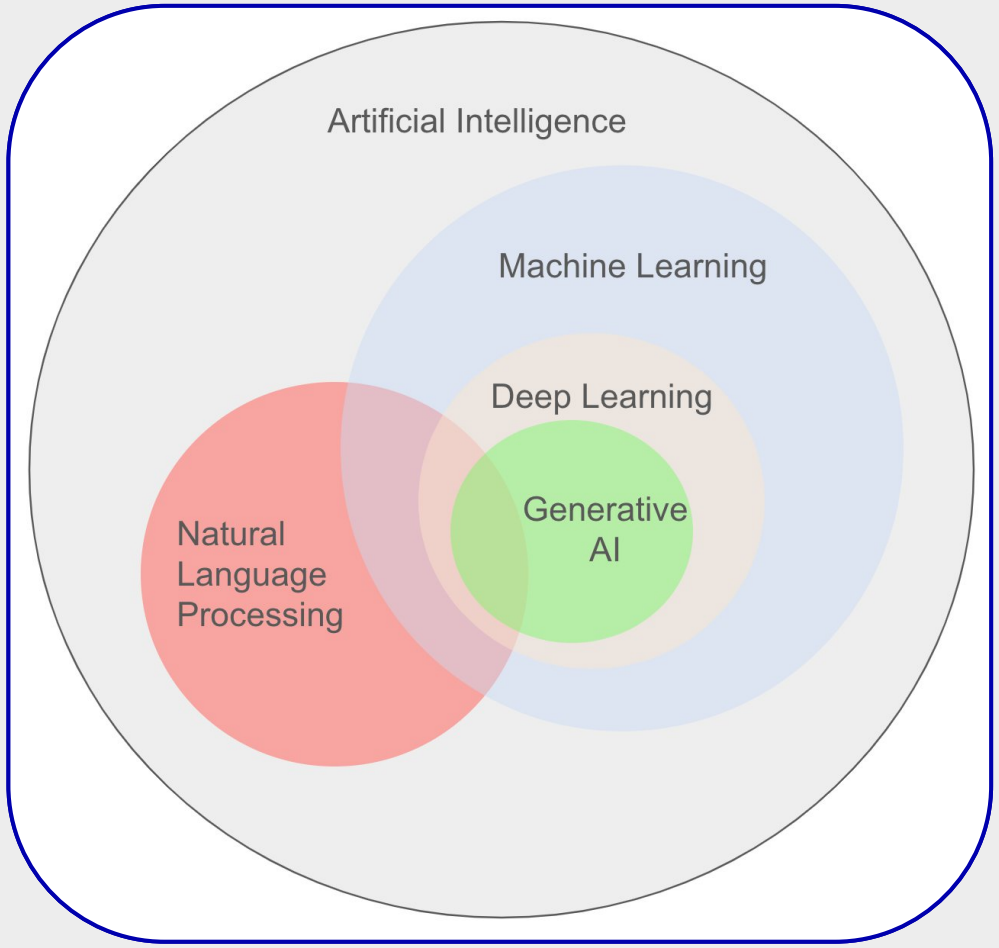
aitho

Overlapping fields in Modern AI

How these areas of AI relate to each other?

Aside is a Venn diagram showing how key AI domains **intersect** and **contain** one another

These overlaps highlight how today's most powerful AI systems combine multiple subfields into a unified capability



aitho

Why LLMs matter?

- Understand **context**, **nuance**, and **intent** in language
- Can be used for **zero-shot** and **few-shot** learning: less data, more generalization
- Enable new capabilities:
  - Conversational AI
  - Semantic search
  - Code generation
  - Document analysis
- Shift from **task-specific ML** to **general-purpose language intelligence**

aitho

LangChain

Managing conversational flows with LLMs

aitho

Introduction to LangChain

What is LangChain?

- Open-source framework for building applications powered by **Large Language Models** (LLMs)
- Facilitates the creation of complex **conversational agents**
- Easily integrates with databases, APIs, external files, and ML models
- Available on **Python** and **JavaScript**

Key components of LangChain

- **Model I/O**
  Standardized interface for interacting with different language models
- **Memory**
  Manages conversational memory for more human-like interactions
- **Chain**
  Modular sequences of model calls (**prompt** → **LLM** → **actions**)
- **Agents**
  Autonomous entities that dynamically decide actions using models

Managing Conversational Flows

- **Memory Types**
  - `BufferMemory` (full conversation history)
  - `SummaryMemory` (automatic summarization)
  - `EntityMemory` (tracks specific entities)
- **Dynamic Routing**
  LangChain can select different conversational flows based on the context

Integrating Language Models

- Supports models like **OpenAI**, **Hugging Face**, **Cohere**, **Anthropic** and more
- **`PromptTemplate`**
  Creates reusable, parameterized prompts

```
prompt_template = PromptTemplate.from_template("Tell me a joke about {topic}")

prompt_template.invoke({"topic": "cats"})
```

- **Model Wrappers**
  Integrate diverse APIs while maintaining a consistent application logic

Use Cases and Benefits

- **Use Cases**
  Smart chatbots, custom search engines, document automation tools
- **Benefits**
  - Modular and flexible
  - Greatly reduces development time
  - Extensible with custom plugins

LangGraph

Building Stateful and Dynamic Conversational Flows

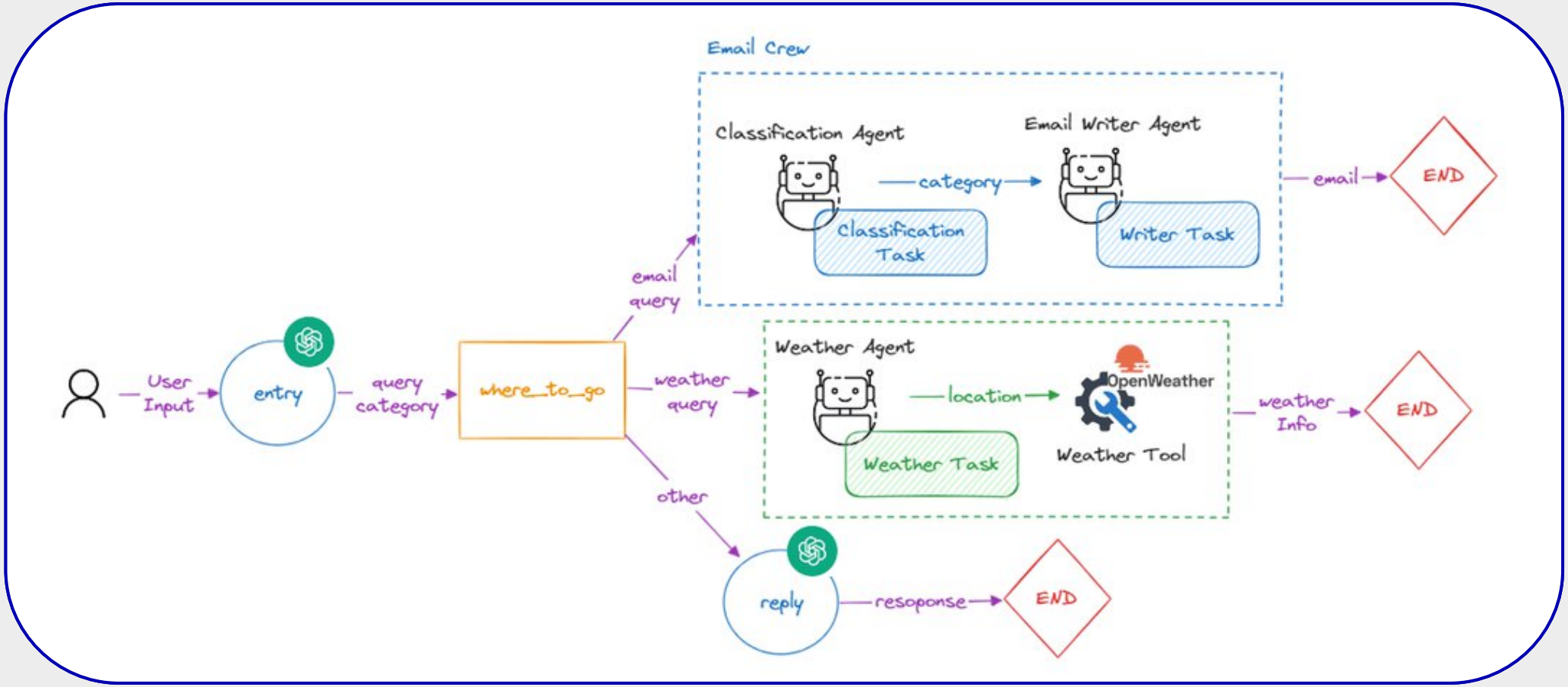
What is LangGraph?

- **LangGraph = LangChain + State Machines + Graph Logic**
  - **Graph-based extension** of LangChain for building complex, dynamic LLM flows
  - Inspired by **finite state machines** and **directed acyclic graphs** (**DAGs**)
  - **Nodes** represent **steps** (e.g. LLM calls, logic), while **edges** define **conditional transitions**
  - Enables **non-linear flows**, loops, and real-time decision branches
- **Why it matters:**
  - Traditional chains are linear
  - LangGraph brings **structure + flexibility** to conversations

LangGraph's Key Features and Use Cases

- **Key Features**
  - **Loop & branching:** perfect for iterative reasoning or fallback logic
  - State tracking: handles memory and user context across steps
  - **Composable: build reusable graph components for complex tasks**
  - Integrated with LangChain tools, memory and agents
- **Use Cases**
  - Multi-step chatbots with error recovery
  - Conversational agents with decision logic
  - Adaptive workflows (e.g. document intake, customer support)

LangGraph: Flow example



Source: ionio.ai

aitho

Let's look at the code!

Thank you!

aitho
welcome home developers

www.aitho.it

partners@aitho.it

Any question?