Beginning LangChain 0.3: A hands-on introduction to essential concepts for LLM-centric applications

Book · July 2025

DOI: 10.5281/zenodo.16529242

1 author:

K. B. Mustapha
University of Nottingham Malaysia Campus
71 PUBLICATIONS **2,366** CITATIONS

SEE PROFILE

Beginning LangChain 0.3 provides a methodical, hands-on introduction to the core packages within LangChain, key classes within the modules of these packages, and important class-specific parameters in these modules. The book emphasises problem-solving strategies.

With each chapter infused with a carefully laid-out scaffolding introduction to build intuition about LangChain concepts and thoughtfully arranged code recipes demonstrating applications, this book will help you get to grips with the core components of LangChain for building GenAI applications.

What to expect:

- Discover how to harness LangChain's prompt template, few-shot template, and chat prompt template to inject prompt engineering techniques into your application logic.
- Experiment with runnables and understand how to leverage them to implement input routing schemes and LLM evaluator workflow
- Gain clarity on how to build a lightweight multi-source RAG workflow that incorporates document processing, vector search, embedding model, and third-party vector stores.
- Learn how to evaluate RAG and how to use evaluation metrics to identify the inadequacy of a RAG pipeline to drive its improvement.
- Understand how to move beyond leaning on LangChain's off-the-shelf evaluator to more advanced evaluation packages, such as DeepEval, in the evaluation procedure
- Work with LangChain's specialized functionalities for interfacing with OpenAI GPT models and HF models

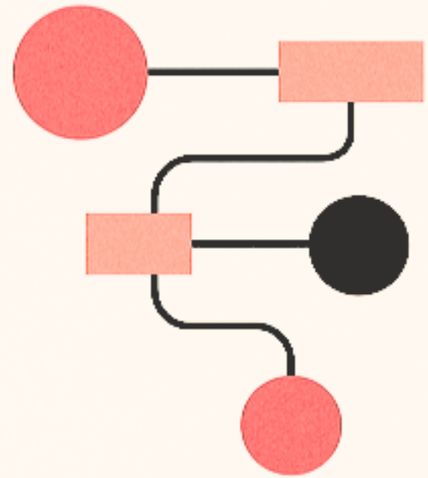
Unique features:

- *LangChain 0.3* – Presents workable code recipes built using the latest LangChain release (v0.3)
- *Model integrations* – Includes examples with OpenAI and Hugging Face-based models.
- *References* – Contains chapters supported by copious references to useful technical blogs and relevant academic articles for further reading and exploration.
- *Code Repo* – Augments with a companion repository for keeping code examples current.

Beginning LangChain 0.3 Mustapha

Beginning LANGCHAIN 0.3

A hands-on introduction to essential concepts for LLM-centric applications



KB Mustapha

Beginning LangChain 0.3

A hands-on introduction to essential concepts for LLM-centric applications

KB MUSTAPHA

Table of Contents

Preface

LangChain has become one of the most prominent frameworks for the development of applications that leverage the power of large language models (LLMs).

A number of factors have contributed to its rise for different LLM-focused applications. On the one hand, the velocity of development around LLMs is staggering. Keeping up with this rapid development on multiple fronts is a challenge for any developer interested in this space. Indeed, every few weeks, new spectacular models are released by different companies (OpenAI, Anthropic, Google, Meta, Cohere). Each model from these vendors may come with its own quirks. Compounding the problem further, developing any sufficiently complex LLM-powered applications requires familiarity with dozens of obscure libraries/packages (across different languages) that predate the ascendance of LLMs. This is where LangChain provides a unique opportunity.

Partly, LangChain addresses the first issue *by offering a standardized abstraction to maneuver the quirks of different models and vendor APIs*. This means you can easily swap out models with minimal changes to your code with a LangChain pipeline.

And with respect to the complexity of developing LLM-powered applications, LangChain does not necessarily eliminate this complexity, *but it helps you to manage it*. With it, you can easily manage prompts, data, and string together components in a modular way.

In short, amid the chaos of rapid evolution in the LLMs space, LangChain features enable developers to focus on building logic and workflows rather than wrestling with low-level API details or tracking every new model release. Ultimately, it is difficult to predict the longevity of its dominance as a framework for LLM-based applications. But as a user of the LangChain ecosystem, I have come to love the system-thinking approach it provides towards steerability of LLMs. Further, I like the vision of the team behind the framework. They have continued to innovate on the ecosystem. It takes a lot of brainpower and experience working in the wild with the underlying technical stacks to be able to come up with some of the abstractions offered by the LangChain ecosystem. In all, with the growing community around the framework shaping its features, it's well-positioned to remain part of the toolkit for LLM-focused software engineering for the foreseeable future.

With the above said, I wrote this book as a reference for my future self, and I hope it helps you too. No doubt, the LangChain's documentation is extensive. I say this humbly, much

of the code in the sprawling documentation may appear lifeless without seeing them in action. This book aims to modestly address this, while bringing together much of the scattered learning I've done on LangChain. Primarily intended for beginners, the book is organized around two main goals:

- To give you a methodical, hands-on introduction to the core packages within the LangChain packages, key classes within the modules in the packages, and important class-specific parameters in these modules.
- To organically introduce you to the foundational knowledge around:
    - Prompt template, few-shot template, chat prompt template, runnables and parsers, and how they are used to compose chains that link various LangChain components with both OpenAI models and open-weight models from the Hugging Face hub.
    - Document processing, embedding, chunking strategy, and vector stores in the context of retrieval-augmented generation (RAG) and how to use evaluation strategies to improve RAG performance.

About Me

This may sound random, but I am an academic who originally specialized in computational mechanics, specifically the finite element method (with three past books on the subject).

Coding has always been my hobby since my PhD days at NTU (Singapore), so when ChatGPT emerged and demonstrated the disruptive power of LLMs, I became fascinated by what this technology means for the broader field of mechanical engineering. I have since been quietly tinkering and employing LLM on a stealth side project. This led me to start learning LangChain. However, I quickly found myself overwhelmed with the documentation, and wish some of the learning materials I used were structured differently. Influenced by this, I decided to write the kind of book I wish I had when I started.

Notably, this book is not written from the perspective of a hardcore programmer, but from that of a curiosity-minded engineer trying to learn and help others learn something genuinely useful. And in writing this book, I am holding on to the old-age adage that the best way to learn something is to teach it. So, I hope the book helps you in your journey to mastery of LangChain.

Audience

This book is for beginners seeking a friendly, code-focused introduction to LangChain. This comprises students, aspiring generative AI engineers, ML engineers, and users interested in crafting generative AI products with LangChain. In style, the book favors practical examples over theory. It is a compendium of recipes that help you build intuition through practical code that spotlights must-know features for getting started with LangChain.

Skill and computing requirements

This book assumes basic knowledge of Python and basic familiarity with large language models (general usage, not the inner workings).

You'll get the most out of the book by running the code snippets as you follow along.

A Python 3.11 (or above) environment is required, and a capable machine is recommended when working with foundation models or large language models. If your local machine isn't powerful enough, consider using cloud platforms like Google Colab or Kaggle to run the examples.

The book's coverage

Put together, the book contains five chapters. A summary of each is provided next.

***Chapter 1*, *Getting started with LangChain*,** presents an overview of LangChain, its benefits (and trade-offs). It further introduces prompt template, chat prompt template, and LangChain expression language (LCEL), and explores how to utilize these to build our first set of modular chains.

Chapter 2*, *Working with chat prompt and few-shot prompt templates, offers a deep dive into the chat prompt and few-shot prompt template. It shows how to work with the chat prompt template to define system-level instructions and how to leverage the few-shot template for static and dynamic examples.

***Chapter 3*, *Building with runnables and output parsers in LangChains*,** focuses on one of the workhorses of LangChain (runnables). Specifically, the chapter covers how to work with runnables to compose robust LangChain components using RunnableLambda,

RunnablePassthrough, RunnableParallel, RunnableSequence, and RunnableBranch. It also demonstrates the idea of using output parsers to get structured output from LLMs.

Chapter 4, Utilizing LangChain facilities for RAG, is dedicated to retrieval-augmented generation (RAG). It provides an overview of RAG and its benefits, RAG strategies, and chunking strategies. It also discusses the integration of LangChain's document loaders, text splitter classes, vector stores, embedding, and retriever classes to build a complete RAG pipeline.

Chapter 5, Performance evaluation in RAG systems, discusses the process of evaluating RAG. It highlights key metrics for measuring RAG system performance and how to use evaluation to identify the weaknesses of the retrieval and generation components of a RAG pipeline. Finally, it highlights how evaluation ties with the idea of evaluation-driven improvement of LLM applications.

Download the example code files

Chapter 1

Getting started with LangChain

This chapter begins with an exploration of the foundational components that give LangChain its power and flexibility: prompt templates, the LangChain Expression Language (LCEL), and chains.

Across sections of the chapters, you will: (i) learn how to combine prompt templates and LCEL to build structured prompts and modular workflows; and (ii) experiment with methods to interact with OpenAI's language models as well as smaller, open-weight models from the Hugging Face Hub.

By the end of this chapter, you will have become familiar with the concept of static and dynamic prompt, be able to compose chains using LCEL, and be able to establish simple interactions with LLMs. In support of the above goal, our discussion will orient around the following topics:

- Introduction to LangChain
- Prompting chat-based models
- Transforming prompts with LangChain templates
- LangChain Expression Language

Requirements

To follow along, you will need:

- Access to a computing environment with Python 3.11 and the associated langchain packages (see sub-section 1.1.4).
- Have or be able to generate an OpenAI key (if you wish to work with OpenAI models) **or** a Hugging Face hub API token (for open-weight models).

1.1 Introduction to LangChain

1.1.1 Overview of the langchain ecosystem

The LangChain ecosystem is broad. Released in 2022, LangChain has become one of the fastest-growing frameworks for the building and scaling of large language model (LLM)-focused applications. The LangChain ecosystem covers the development, deployment, and monitoring of LLM-powered applications as depicted in Figure 1.1.

Constituting the development layers are various interlinked **LangChain packages and components**. These packages and components provide standardized interfaces for working with models, prompts, memory, and tools.

Similarly, **LangGraph** is an offshoot of the LangChain ecosystem that focuses on the building of LLM-based agentic applications using a graph-based paradigm. So far, this graph-based approach to building agents has received favorable adoption for designing smart workflows for agentic systems.

On the observability side, **LangSmith** enables logging of LangChain's application processes, testing (e.g., A/B testing), and debugging.

Overall, each layer of LangChain's ecosystem is designed to plug into others easily. Together, these functionalities ease the iteration cycle in the development of LLM workflows. They also provide developers with the facility for scaling prototypes to production systems.

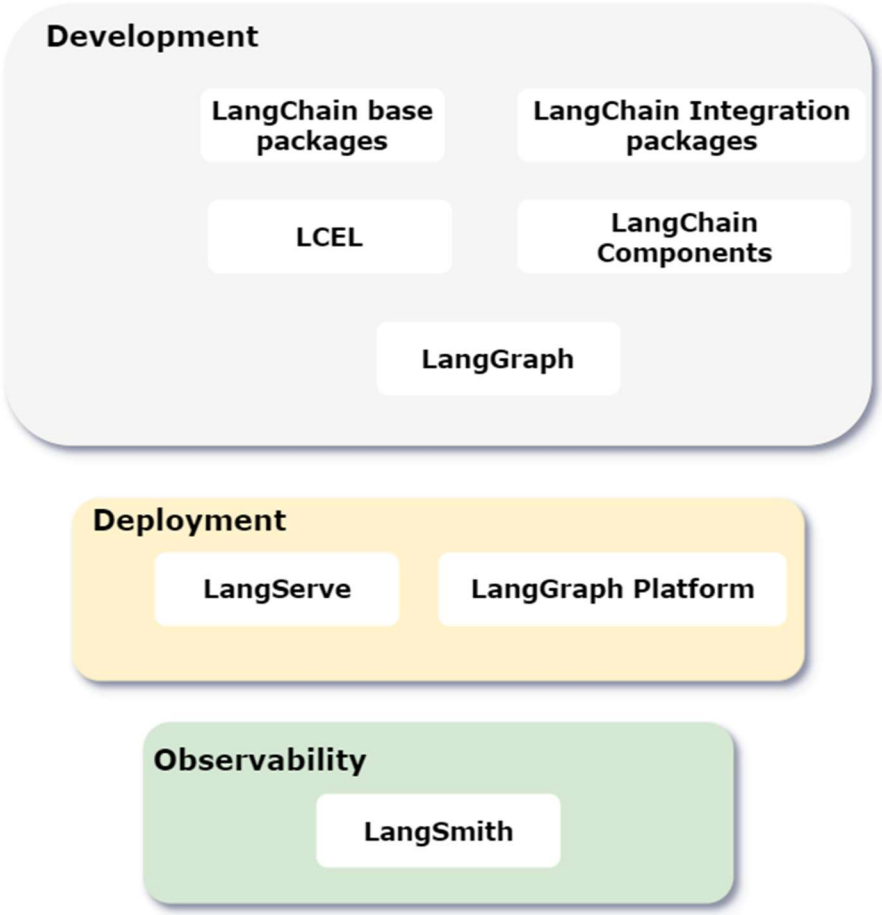


Figure 1.1: The LangChain ecosystem

1.1.2 Benefits (and trade-offs) of using LangChain

LangChain is one of the most popular integration libraries for LLM. Its popularity is evident in several surveys. For instance, the 2025 *State of AI* report highlights it as a leading tool for LLM adoption (as shown in Figure 1.2) [1].

Similarly, a recent survey of LLM developers echoes the same trend [2]. A crucial factor in the strength of LangChain's rise to prominence is its ability to liberate developers from the labyrinth of small but necessary libraries needed for building LLM-based applications.