Watsonx.ai

Introduction to LangChain

Elena Lowery (elowery@us.ibm.com)
Global Sales
Andre de Waal (andre.dewaal@us.ibm.com)
Curriculum Development

watsonx

IBM

Agenda

- Large language model use cases
- LangChain
- A simple chain
- Example
- Features
- How to access LangChain
- Hands-on Lab
- References

Large language model use cases

1. Document question answering
2. Chatbots
3. Analyzing structured data
4. Generative question answering
5. Retrieval-augmented generation
6. Summarization
7. Personal assistants
8. Interacting with APIs
9. Extraction
10. Code generation
11. Many more ...

Task:

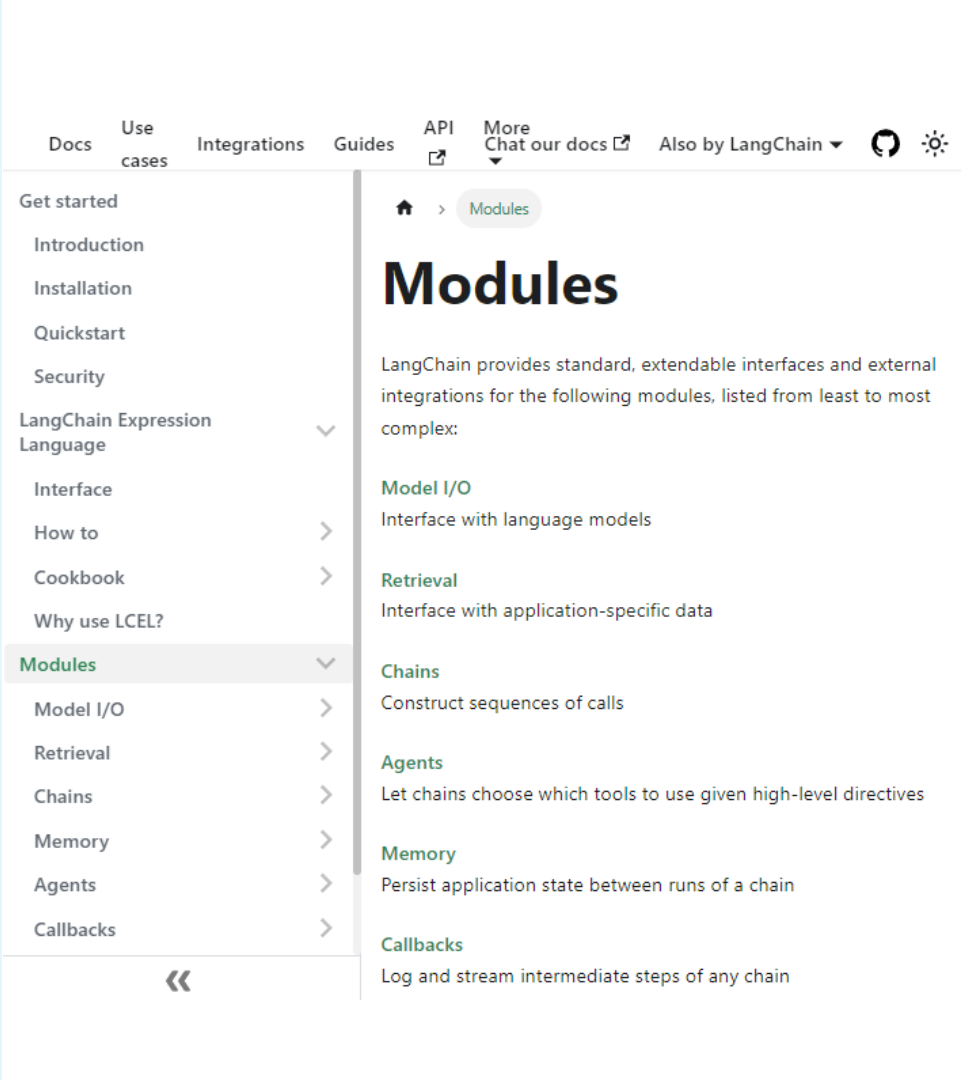
You need to develop AI-powered applications where large language models interact with other large language models, applications and tools. How do you simplify the creation of these applications?

LangChain

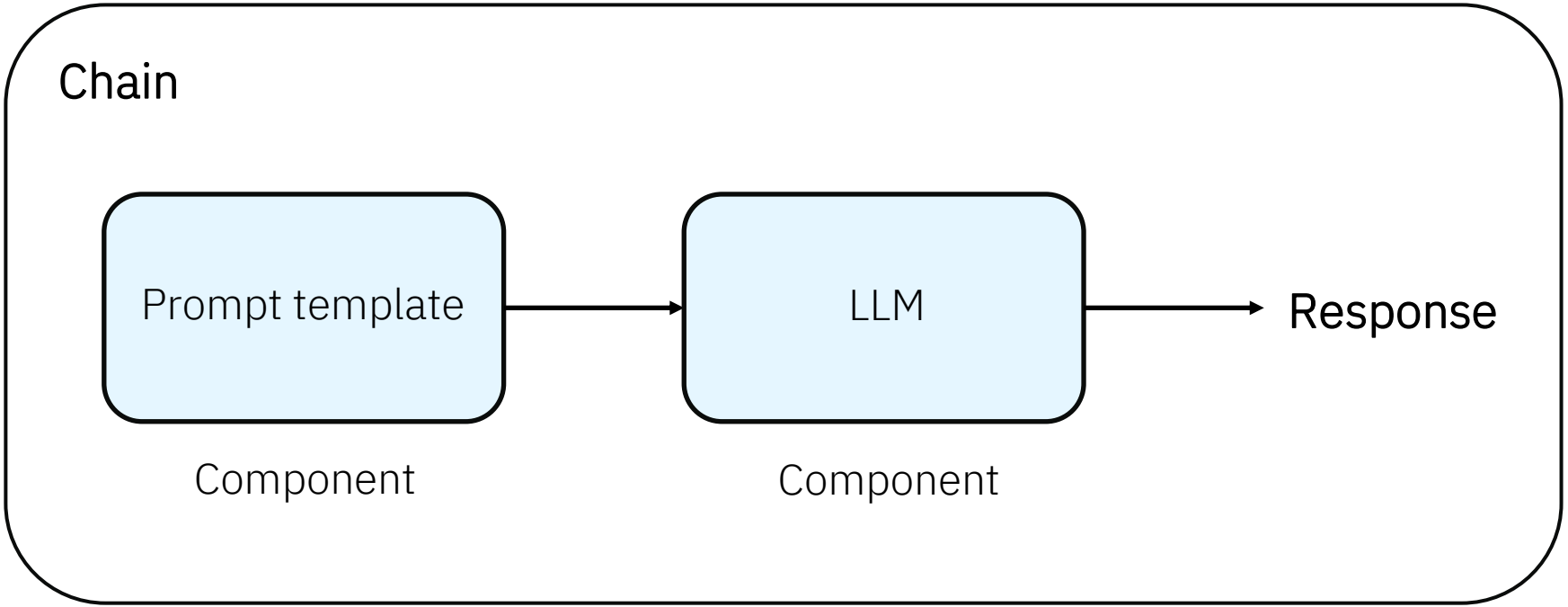
1. LangChain is a popular open source framework that's becoming a de facto standard for LLM application development.

2. LangChain simplifies implementation of many tasks that are typical in LLM applications and helps us write "cleaner" code.

3. LangChain does not add new capabilities to LLMs.

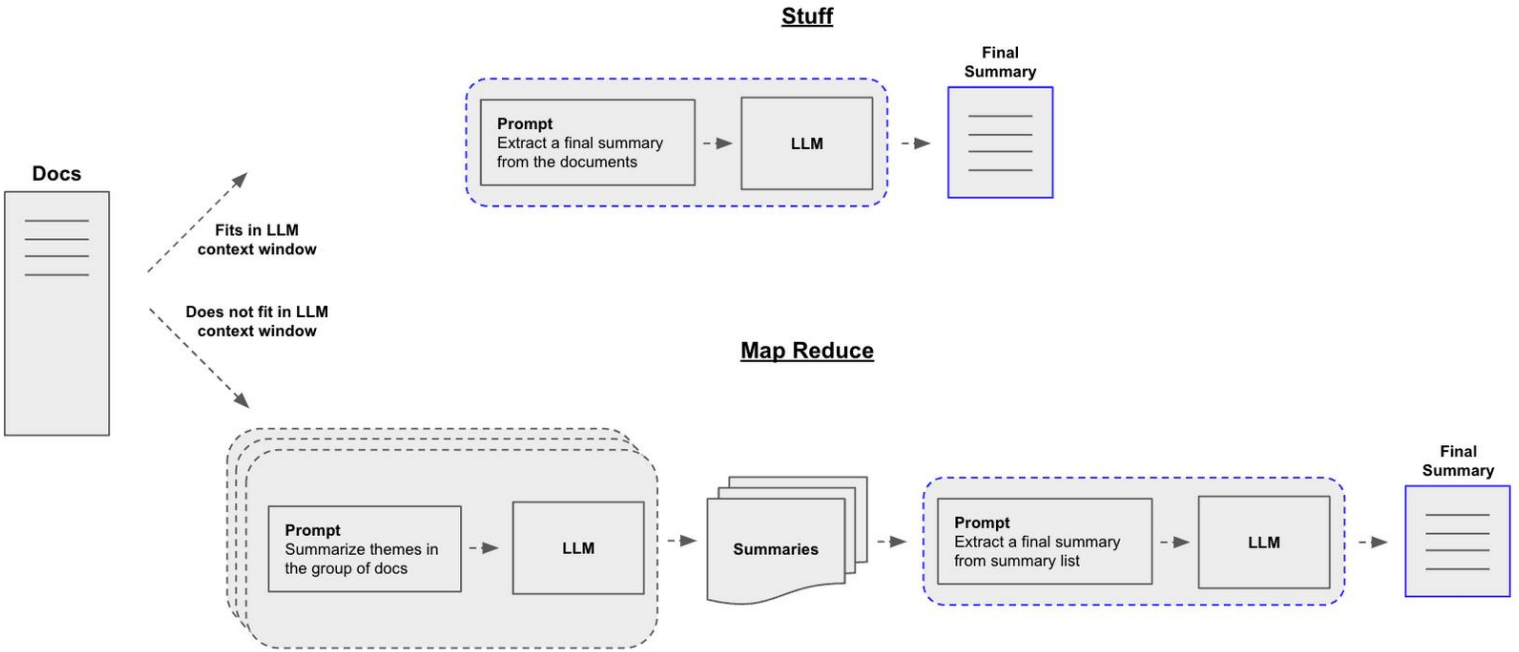


A simple chain



A simple LLM chain, where a prompt template component is passed to an LLM component to get a response from the language model.

Example



- This typical summarization use case requires a lot of “orchestration” and “utility” code.
- LangChain provides an API to simplify implementation.

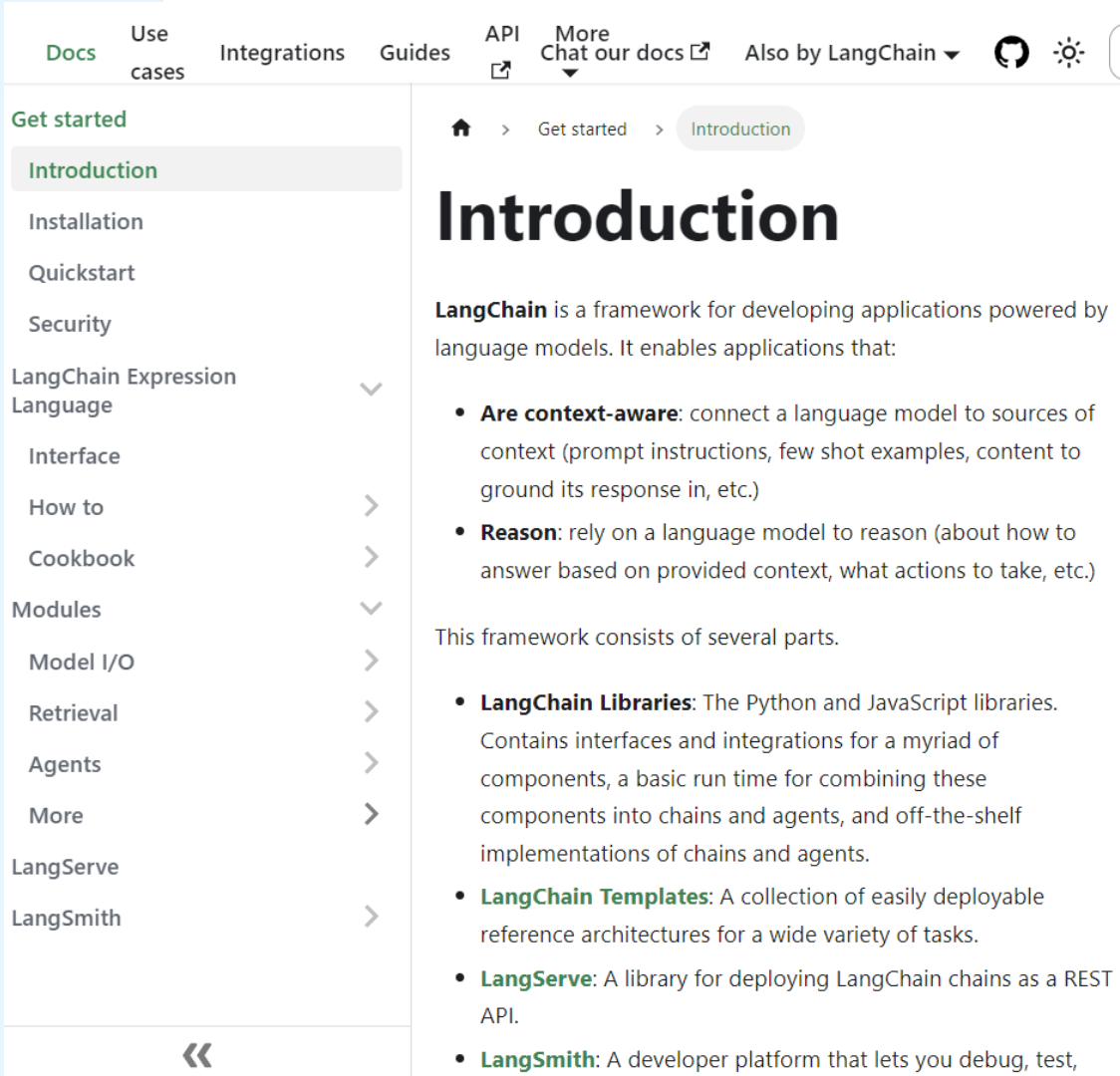
Features

LangChain is:

- A framework for developing applications powered by language models
- A feature-rich, fast-growing
- API

Popular “utility” features are:

- Prompt templates
- LLM output parsers
- Wrappers around a series of single components (chains)
- Maintaining session state between LLM calls (memory)
- Support for RAG (specific pattern)



Features

- LangChain can work with LLMs developed by different vendors, as most LLMs provide a “generic API”
- At IBM, LangChain support and integration for a specific feature needs to be officially supported by IBM Watson Machine Learning REST API (see documentation)

Extensions

LangChain

class ibm_watson_machine_learning.foundation_models.extensions.langchain.**WatsonxLLM**(**model**) [source]

LangChain CustomLLM wrapper for watsonx foundation models.

PARAMETERS:

model (*Model*) – foundation model inference object instance

Supported chain types:

- *LLMChain*,
- *TransformChain*,
- *SequentialChain*,
- *SimpleSequentialChain*
- *ConversationChain* (including *ConversationBufferMemory*)
- *LLMMathChain* (`bigscience/mt0-xxl`, `eleutherai/gpt-neox-20b`, `ibm/mpt-7b-instruct2`, `bigcode/starcoder`, `meta-llama/llama-2-70b-chat`, `ibm/granite-13b-instruct-v1` models only)

How to access LangChain

From IBM Cloud:

- Create a Watson Studio Project
- Create and associate a Watson Machine (WML) service with the Project
- Install ibm-watson-machine-learning Python library
- Define WML credentials
- Install langchain

From a Python Integrated Development Environment (IDE) such as PyCharm:

- Create a Watson Studio Project with associated WML service in the IBM Cloud
- Install ibm-watson-machine-learning Python library in IDE
- Define WML credentials
- Install langchain in IDE

Installation

The *ibm-watson-machine-learning* Python library is publicly available on PyPI: https://pypi.org/project/ibm-watson-machine-learning/.

The package can be installed with pip:

```
$ pip install ibm-watson-machine-learning
```

Note

Additionally, the *ibm-watson-machine-learning* Python library is integrated into the Watson Studio Jupyter notebook.

Product Offerings

The python package supports the following product offerings:

- IBM Cloud Pak for Data as a Service
- IBM Cloud Pak for Data:
    - Version 3.5
    - Version 4.0
    - Version 4.5
    - Version 4.6
    - Version 4.7

Hands-on lab

- Implement your own sequential chain
- Implement memory with LangChain
- Implement a simple RAG use case with LangChain

References

- API - IBM Watson Machine Learning (IBM documentation)
- Extensions - IBM Watson Machine Learning (IBM documentation)
- DLAI - Learning Platform (deeplearning.ai) (course)
- Introduction | 🦜🔗 Langchain (documentation)
- Community navigator | 🦜🔗 Langchain (community)
- Introduction to LangChain: A Framework for LLM Powered Applications (davidgentile.net) (blog)