

Basic Facts on LangChain

Alexander C. Mueller

July 2023

Agenda

Goal: answer basic, standard questions about LangChain

- What is it?
- Who is it for?
- What problems does it solve?

Useful Tools and Python Examples

- Templates
- Ingesting documents
- The Chain interface

Relevant businesses and people

What is it?

“LangChain is a framework designed to simplify the creation of applications using large language models (LLMs).”

- Wikipedia
- GitHub
- langchain.com
- Versions in Python (and Javascript)
- Released under an MIT license

I will focus on the Python version.

Who is it for?

Developers working in Python (or JavaScript) can use it to manage their interactions with OpenAI's APIs and similar.

From the Python Quickstart

Installation

To install LangChain run:

Pip Conda

```
pip install langchain
```

For more details, see our [Installation guide](#).

Environment setup

Using LangChain will usually require integrations with one or more model providers, data stores, APIs, etc. For this example, we'll use OpenAI's model APIs.

First we'll need to install their Python package:

```
pip install openai
```

What problem does it solve?

You have a lot of complicated parameters, etc. and ancillary tasks around using this LLM (or LLMs!) that you would like to manage in one place.

```
from langchain.chat_models import ChatOpenAI
from langchain.schema import (
    AIMessage,
    HumanMessage,
    SystemMessage
)

chat = ChatOpenAI(temperature=0)
chat.predict_messages([HumanMessage(content="Translate this sentence from English to French. I love programming.")])
# >> AIMessage(content="'J'aime programmer.", additional_kwargs={})
```

Example: Prompt Templates

From this page in the Python docs

```
from langchain import PromptTemplate

template = """
You are a naming consultant for new companies.
What is a good name for a company that makes {product}?
"""

prompt = PromptTemplate.from_template(template)
prompt.format(product="colorful socks")
```

Few-shot templates are a more interesting example.

Example: Ingesting Data

From [this page](#) in the Python docs

```
from langchain.document_loaders.csv_loader import CSVLoader
```

```
loader = CSVLoader(file_path='./example_data/mlb_teams_2012.csv')  
data = loader.load()
```

```
print(data)
```

```
[Document(page_content='Team: Nationals\n"Payroll (millions)": 81.34\n"Wins": 98', lookup_str='', metad
```

You can similarly load pdfs and lots of other common sorts of files.

The Chain Interface

From [this page](#) in the Python docs.

“Using an LLM in isolation is fine for simple applications, but more complex applications require chaining LLMs - either with each other or with other components.”

```
from langchain.llms import OpenAI
from langchain.prompts import PromptTemplate

llm = OpenAI(temperature=0.9)
prompt = PromptTemplate(
    input_variables=["product"],
    template="What is a good name for a company that makes {product}?",
)
```

We can now create a very simple chain that will take user input, format the prompt with it, and then send it to the LLM.

```
from langchain.chains import LLMChain
chain = LLMChain(llm=llm, prompt=prompt)

# Run the chain only specifying the input variable.
print(chain.run("colorful socks"))
```


Relevant People and Businesses

Created by Harrison Chase.

Affiliated with LangChain, Inc. (langchain.com)

- Significant VC Seed Funding
- Based in Singapore
- Maybe a bunch of employees?

They seem to be pretty focused on Discord.