

LangChain and LLMs: The Future of AI-Generated Content

Ivan Reznikov
@qburst @mdxdubai

Maksym Nechepurenko
@agility

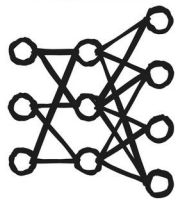
Plan

- What is LangChain?
- What are the components of LangChain?
- How do LangChain components work together?
- Practical cases
- Future for AI-generated content

What is LangChain?

LangChain is a framework that makes it easier to create applications using large language models (LLMs).

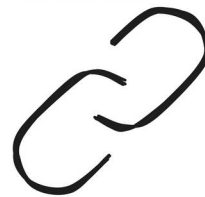
Models



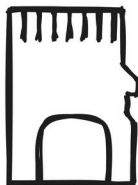
Prompts



Chains



Memory



Indexes



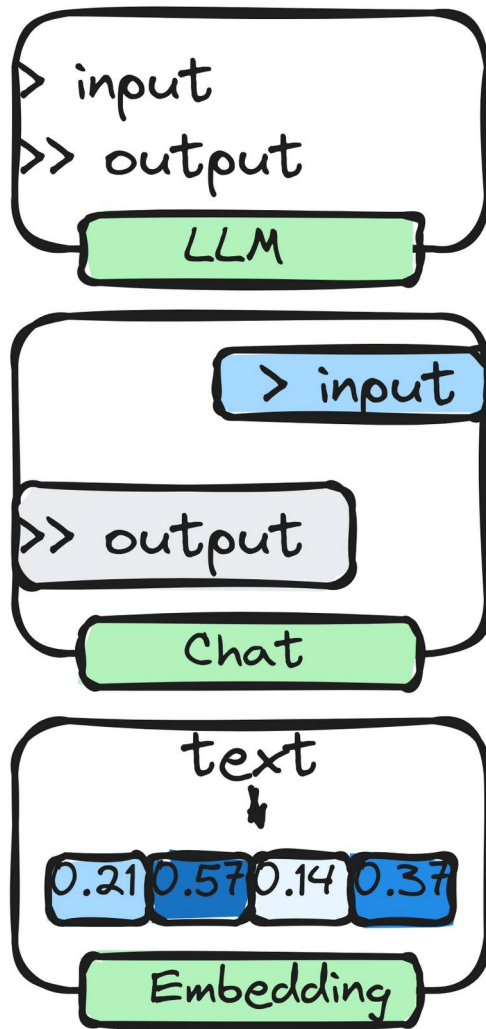
Agents and Tools



Models

LangChain supports a variety of different models, so you can choose the one that suits your needs best

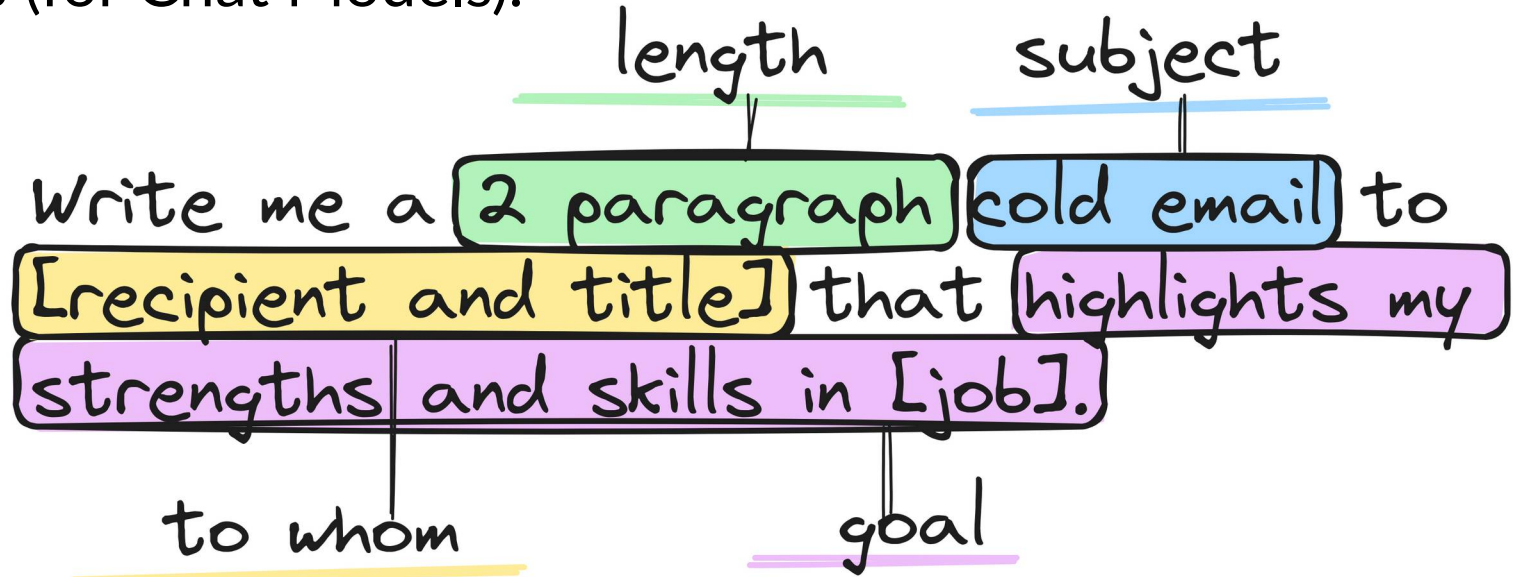
- **Language Models** are used for generating text
 - LLMs utilize APIs that take input text and generate text outputs
 - ChatModels employ models that process chat messages and produce responses.
- **Text Embedding Models** convert text into numerical representations



Prompts

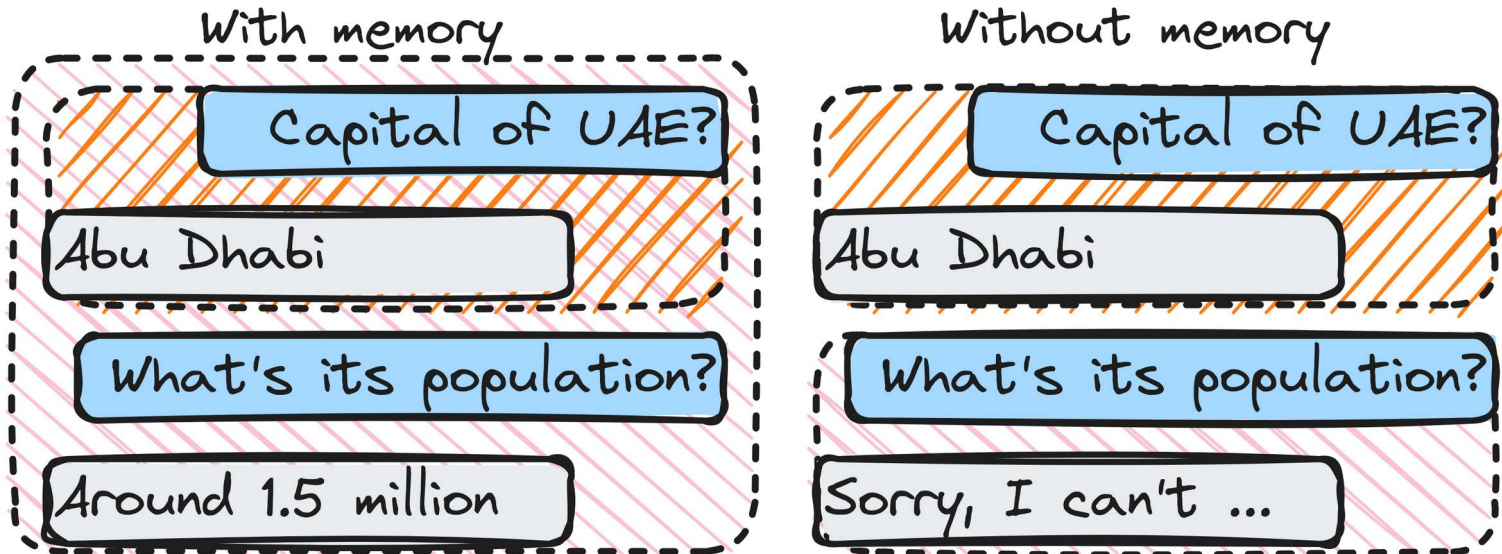
Prompts are the instructions that you give to the LLM. They tell the LLM what you want it to do, and how you want it to do it.

Prompts can take the form of a string (for Language Models) or a list of messages (for Chat Models).



Memory

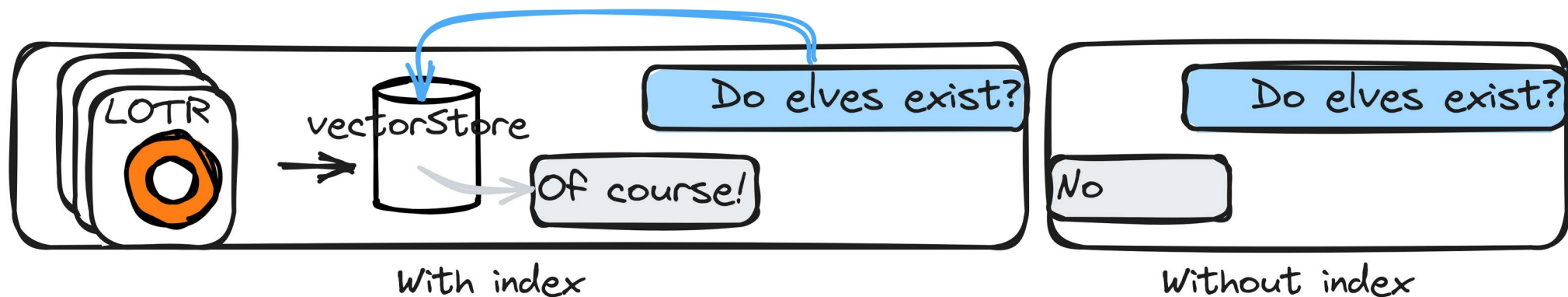
LangChain introduces memory components that enable the retention and utilization of past interactions. By default, LangChain components are stateless, treating each query independently. Memory in LangChain allows for the storage and management of previous chat messages.



Indexes

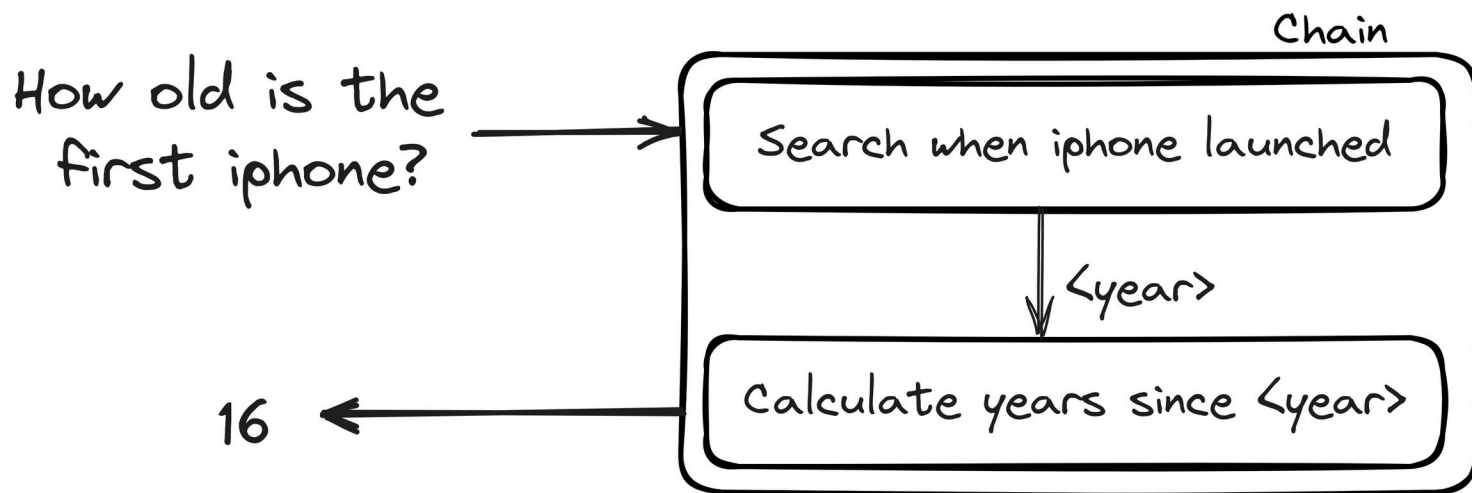
Indexes are databases of information that can be used to provide context to the LLM.

For example, if you are asking the LLM to answer a question about a particular topic, you could use an index to provide the LLM with a list of relevant articles or websites.



Chains

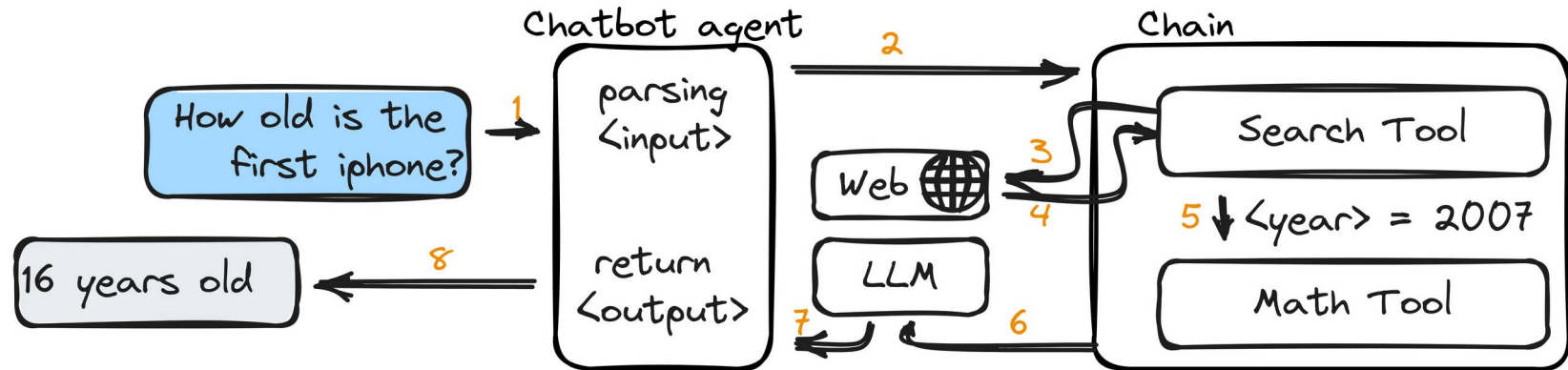
Chains are sequences of calls to the LLM. They allow you to perform complex tasks by chaining together multiple calls to the LLM. There are multiple types of chains to solve different cases.



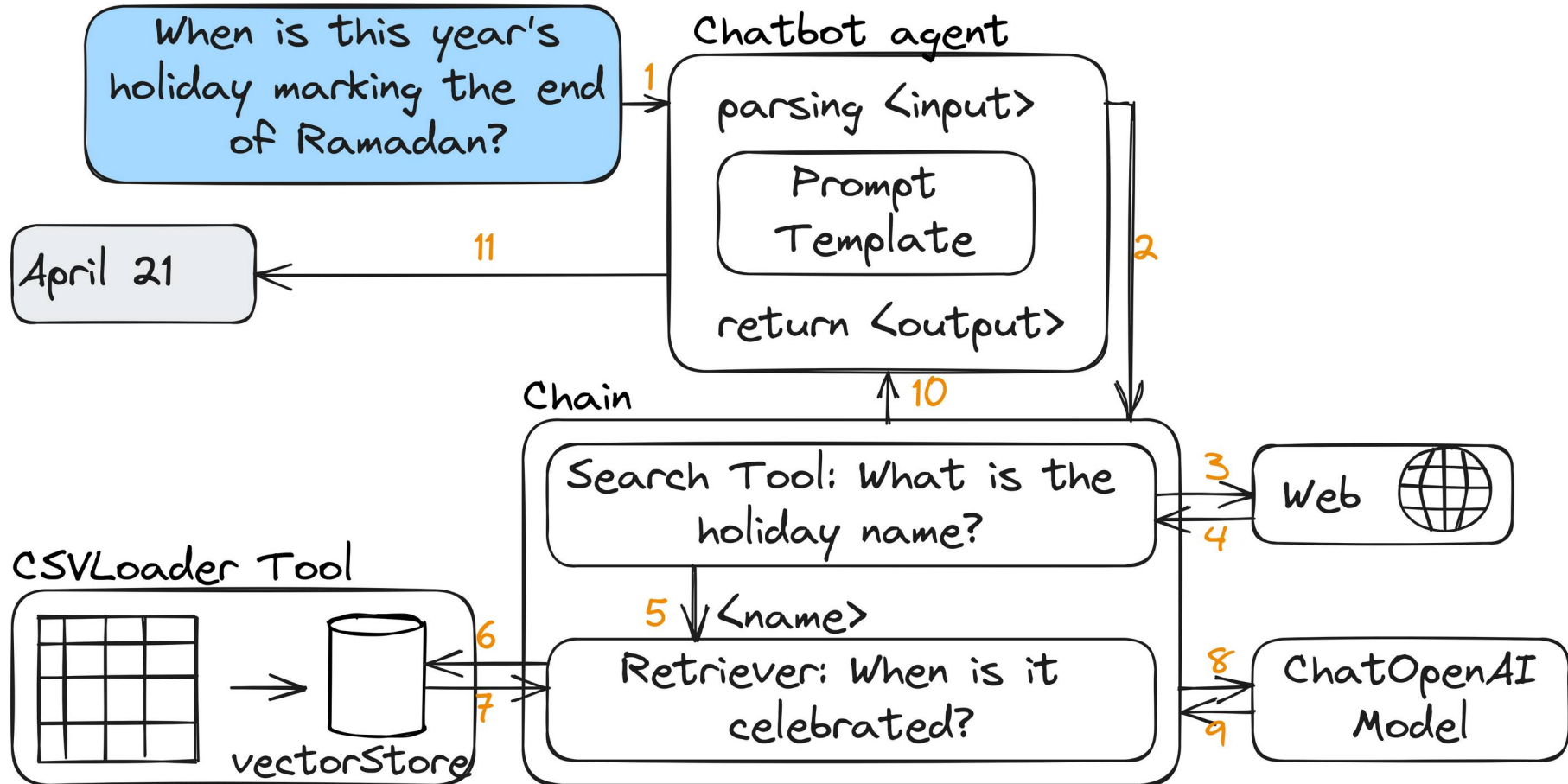
Agents and Tools

Agents are objects that manage interaction between the user and the LLM. They handle things like *parsing input*, *generating output* and determining the sequence of actions to follow and tools to use.

A tool is a function designed to perform a specific task. Examples of tools can range from *Google Search*, *DB lookups*, *API calls*, to other chains. The standard interface for a tool is a function that accepts a string as input and returns a string as output.



Scenario: Public Holidays



Future of AI-generated content

Nobody can know for sure.

My guts tell me, that ai-generated content will soon be the new norm. And we'll adapt.

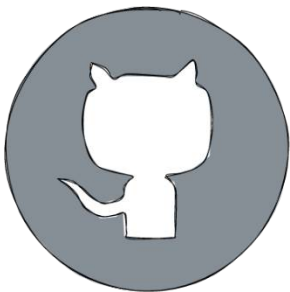
AI is great at connecting the dots. And using computational powers, we'll be able to reduce the manual part of it.

Skynet isn't here. Yet. Langchain and LLMs are Excel on ML steroids.

Time for questions



<https://www.linkedin.com/in/reznikovivan/>



<https://github.com/IvanReznikov>



ivanreznikov@gmail.com