Welcome to LlamaIndex 🦙 !

LlamaIndex is the leading framework for building LLM-powered agents over your data with LLMs and workflows.

Introduction

What is context augmentation? What are agents and workflows? How does LlamaIndex help build them?

Use cases

What kind of apps can you build with LlamaIndex? Who should use it?

Getting started

Get started in Python or TypeScript in just 5 lines of code!

LlamaCloud

Managed services for LlamaIndex including LlamaParse, the world's best document parser.

Community

Get help and meet collaborators on Discord, Twitter, LinkedIn, and learn how to contribute to the project.

Related projects

Check out our library of connectors, readers, and other integrations at LlamaHub as well as demos and starter apps like create-llama.

Introduction

What are agents?

Agents are LLM-powered knowledge assistants that use tools to perform tasks like research, data extraction, and more. Agents range from simple question-answering to being able to sense, decide and take actions in order to complete tasks.

LlamaIndex provides a framework for building agents including the ability to use RAG pipelines as one of many tools to complete a task.

What are workflows?

Workflows are multi-step processes that combine one or more agents, data connectors, and other tools to complete a task. They are event-driven software that allows you to combine RAG data sources and multiple agents to create a complex application that can perform a wide variety of tasks with reflection, error-correction, and other hallmarks of advanced LLM applications. You can then deploy these agentic workflows as production microservices.

What is context augmentation?

LLMs offer a natural language interface between humans and data. LLMs come pre-trained on huge amounts of publicly available data, but they are not trained on **your** data. Your data may be private or specific to the problem you're trying to solve. It's behind APIs, in SQL databases, or trapped in PDFs and slide decks.

Context augmentation makes your data available to the LLM to solve the problem at hand. LlamaIndex provides the tools to build any of context-augmentation use case, from prototype to production. Our tools allow you to ingest, parse, index and process your data and quickly implement complex query workflows combining data access with LLM prompting.

The most popular example of context-augmentation is Retrieval-Augmented Generation or RAG, which combines context with LLMs at inference time.

LlamaIndex is the framework for Context-Augmented LLM Applications

LlamaIndex imposes no restriction on how you use LLMs. You can use LLMs as auto-complete, chatbots, agents, and more. It just makes using them easier. We provide tools like:

- **Data connectors** ingest your existing data from their native source and format. These could be APIs, PDFs, SQL, and (much) more.
- **Data indexes** structure your data in intermediate representations that are easy and performant for LLMs to consume.
- **Engines** provide natural language access to your data. For example:
  - Query engines are powerful interfaces for question-answering (e.g. a RAG flow).
  - Chat engines are conversational interfaces for multi-message, "back and forth" interactions with your data.
- **Agents** are LLM-powered knowledge workers augmented by tools, from simple helper functions to API integrations and more.
- **Observability/Evaluation** integrations that enable you to rigorously experiment, evaluate, and monitor your app in a virtuous cycle.
- **Workflows** allow you to combine all of the above into an event-driven system far more flexible than other, graph-based approaches.

Use cases

Some popular use cases for LlamaIndex and context augmentation in general include:

- Question-Answering (Retrieval-Augmented Generation aka RAG)
- Chatbots
- Document Understanding and Data Extraction
- Autonomous Agents that can perform research and take actions
- Multi-modal applications that combine text, images, and other data types
- Fine-tuning models on data to improve performance

Check out our use cases documentation for more examples and links to tutorials.

Who is LlamaIndex for?

LlamaIndex provides tools for beginners, advanced users, and everyone in between.

Our high-level API allows beginner users to use LlamaIndex to ingest and query their data in 5 lines of code.

For more complex applications, our lower-level APIs allow advanced users to customize and extend any module -- data connectors, indices, retrievers, query engines, and reranking modules -- to fit their needs.

Getting Started

LlamaIndex is available in Python (these docs) and Typescript. If you're not sure where to start, we recommend reading how to read these docs which will point you to the right place based on your experience level.

30 second quickstart

Set an environment variable called `OPENAI_API_KEY` with an OpenAI API key. Install the Python library:

```
pip install llama-index
```

Put some documents in a folder called `data`, then ask questions about them with our famous 5-line starter:

```
from llama_index.core import VectorStoreIndex, SimpleDirectoryReader

documents = SimpleDirectoryReader("data").load_data()
index = VectorStoreIndex.from_documents(documents)
query_engine = index.as_query_engine()
response = query_engine.query("Some question about the data should go here")
print(response)
```

If any part of this trips you up, don't worry! Check out our more comprehensive starter tutorials using remote APIs like OpenAI or any model that runs on your laptop.

LlamaCloud

If you're an enterprise developer, check out **LlamaCloud**. It is an end-to-end managed service for data parsing, ingestion, indexing, and retrieval, allowing you to get production-quality data for your production LLM application. It's available both hosted on our servers or as a self-hosted solution.

LlamaParse

LlamaParse is our state-of-the-art document parsing solution. It's available as part of LlamaCloud and also available as a self-serve API. You can sign up and parse up to 1000 pages/day for free, or enter a credit card for unlimited parsing. Learn more.

Community

Need help? Have a feature suggestion? Join the LlamaIndex community:

- Twitter
- Discord
- LinkedIn

Getting the library

- LlamaIndex Python
  - LlamaIndex Python Github
  - Python Docs (what you're reading now)
  - LlamaIndex on PyPi
- LlamaIndex.TS (Typescript/Javascript package):
  - LlamaIndex.TS Github
  - TypeScript Docs
  - LlamaIndex.TS on npm

Contributing

We are open-source and always welcome contributions to the project! Check out our contributing guide for full details on how to extend the core library or add an integration to a third party like an LLM, a vector store, an agent tool and more.

LlamaIndex Ecosystem

There's more to the LlamaIndex universe! Check out some of our other projects:

- llama_deploy | Deploy your agentic workflows as production microservices
- LlamaHub | A large (and growing!) collection of custom data connectors
- SEC Insights | A LlamaIndex-powered application for financial research
- create-llama | A CLI tool to quickly scaffold LlamaIndex projects