LangStream

A short technical demo

LangStream

langstream.ai

github.com/LangStream

Framework for building event-driven GenAI applications, across LLMs and vector databases, no-code / low-code

Generative AI, with all the things!

LangStream integrates with LLMs from OpenAI, Google Vertex AI, and Hugging Face

Vector databases like Pinecone and DataStax, Python libraries like LangChain and LlamaIndex.

Lets you build Q&A chat over unstructured text in minutes

Works with your embeddings in Vector databases, automatically create and update embeddings in your vector database for unstructured data (PDFs, Word documents, HTML) using proprietary and open-source embeddings models.

Easy development in Visual Studio, deploy the same things to Dev or Prod!

Deployment

Define the Pipeline for your Application, configuring the built-in “Agents”, then deploy to cloud-native microservices joined together with fault-tolerant events

Deploy the same Application the same way everywhere, for Dev and Prod

Dev - local docker run, bundling everything you need to run and test

Test - minikube deploys to a local k8s, for integration testing

Prod - cloud-native event-driven architecture, kubernetes with helm

That’s a lot of buzzwords…

But does it work?

Is it hard to get started?

Is it worth the effort?

What’s wrong with a load of python scripts on someone’s laptop anyway?

Demo time!

Demo 1 - Explanations!!

github.com/LangStream/langstream/tree/main/examples/applications/python/python-processor-exclamation

3 lines of Python to add !! to everything sent through the pipeline

12 lines of YAML configuration to build the Application pipeline

Local “docker run” to deploy the Application to our local dev machine

Demo chat interface to let us send + receive event messages to test with

Demo 2 - OpenAI API for Chat

github.com/LangStream/langstream/tree/main/examples/applications/openai-text-completions

Use your OpenAI developer API token to talk to their GPT-3.5 model

Sends messages to their Chat API, and streams back the results

0 lines of code needed! ~20 lines of YAML

Again, local docker deployment (API key via env), access via chat interface

Demo 3 - Hugging Face Embeddings

OpenAI and Google are pretty good, but they’re not free

HuggingFace has a huge number of excellent open models

LangStream works with both the HF API, and on-demand model downloads

github.com/LangStream/langstream/tree/main/examples/applications/compute-hugging-face-embeddings

Specify which HF embedding model via Environment variable, YAML pipeline to compute the embeddings.

Would normally then feed into a Vector Database, for eg RAG chatbot

Embeddings?

That’s a whole different talk!

Multi-dimensional vector space, where similar things are near

Helps you turn your text (or documents, or images, or sound) into a series of numbers, which the AI can then work on

Want to learn more?

https://simonwillison.net/2023/Oct/23/embeddings/

https://ig.ft.com/generative-ai/

Demo 4 - Poor man’s Vector DB with Python

Use a model from HuggingFace to compute the embeddings

Pick a multi-lingual model, so that similar words from different languages get similar embeddings

Few lines of Python numpy to compute the cosine similarity between embeddings

Not intended for production!

25 lines of Python, 25 lines of YAML, if you need custom code it isn’t hard

Come join us!

https://github.com/LangStream/

https://langstream.ai/

https://langstream.slack.com/

@langstream_ai

Extras!

Please note - the following demos are not yet available in LangStream

They are based on the same components that LangStream uses, but haven’t been integrated (yet)

We’re hoping to give you an idea of what’s possible in future, we can’t promise when (if!) they’ll be available

But they’re pretty amazing…

Image Classification

clip.cpp + openclip supports zero-shot classification of images

pass in an image and a list of possible classifications, get back scores

eg if it contains people, send it to a different pipeline

eg if it contains lots of text, run OCR on it

eg decide what kind of animal it is, to prompt text retrieval of articles

Image Search

clip.cpp + openclip supports generating embeddings for both images and text, in the same vector space

1. Index all your images into your vector database
2. Prompt the user for what they are looking for, generate embedding of that text, query vector database, show top images
3. Bit like RAG case, but querying images not text
4. Or could even do both images and text!

eg search for “raccoon”, find all images containing one
https://simonwillison.net/2023/Sep/12/llm-clip-and-chat/

Image Description / Captions / OCR

llama.cpp + llava-1.5 supports "chat for images"

- eg "please describe this image in detail"
- eg "does this image contain any people?"
- eg "read the text from this image"

Model is much larger than the OpenCLIP one, so a lot slower to execute inference

But LangStream’s event-driven asynchronous architecture means that isn’t always an issue for use!