LangChain 0.0.312

LangChain	LLM application development framework	
	Systeemvereisten:	Python, JS
	Ontwikkeld door:	LangChain
MIT licentie	Contactpersoon:	Bert.Vanhalst@smals.be

Functionaliteiten

LangChain is een open source framework voor het bouwen van toepassingen op basis van taalmodellen (*large language models* - LLM's). Het laat toe om op een eenvoudige manier de verschillende componenten van dergelijke toepassingen te orchestreren op basis van *chains*. Het framework bestaat uit verschillende onderdelen:

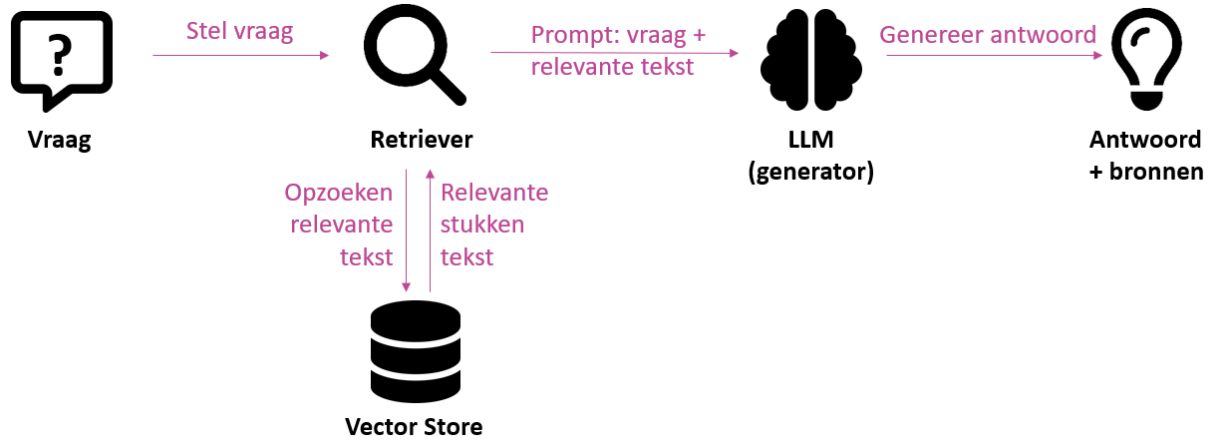
- LangChain Libraries: Python en Javascript libraries met interfaces en integraties voor verschillende componenten, waaronder:
  - Document loaders voor het inladen van content uit verschillende bronnen, met ondersteuning voor verschillende formaten zoals PDF en HTML; integraties met third party tools voor het inladen van content;
  - Text splitters voor het opdelen van grote documenten in kleinere stukken, wat nuttig is bij het verwerken van grote tekstdocumenten die de token limiet van taalmodellen overschrijden;
  - Integraties met verschillende LLM's, chat models en text embedding models zoals OpenAI, Azure OpenAI, etc;
  - Integraties met talloze vector store providers zoals Chroma, Pinecone, Weaviate, etc;
- LangChain Templates: een verzameling van referentiearchitecturen die gemakkelijk kunnen gedeployed worden met LangServe (zie hieronder);
- LangServe: een library voor het deployen van chains als REST API;
- LangSmith: een platform voor het debuggen, testen, evalueren en monitoren van chains die gebouwd zijn met gelijk welk LLM framework, met naadloze integratie met LangChain.

Conclusies & Aanbevelingen

LangChain is een krachtig framework voor het bouwen van toepassingen op basis van taalmodellen. Het is nog jong (één jaar) en in volle evolutie waardoor documentatie frequent wijzigt. Maar het wordt sterk ondersteund door een community van meer dan 2000 ontwikkelaars die bijdragen.

Testen & Resultaten

Een typische use case voor LLM's is *question anwering* op basis van een zogeheten RAG-architectuur (*Retrieval Augmented Generation*, zie schema hieronder), waarbij een vraag in natuurlijke taal beantwoord wordt door een taalmodel (LLM) op basis van de meest relevante stukken tekst uit een knowledge base.



Als test bouwden we een dergelijke question answering toepassing op basis van een PDF-bestand met de RSZ-instructies aan de werkgevers. We schakelden LangChain in voor het voorbereiden van de index (data ingestion) en als orchestrator om de verschillende componenten van het systeem aan elkaar te koppelen tijdens de uitvoeringsfase voor het beantwoorden van vragen.

Ingestion: Naast het inlezen van het PDF-bestand zijn er functies voor het opsplitsen van de tekst in kleinere stukken (*MarkdownHeaderTextSplitter* en *RecursiveCharacterTextSplitter*). Dit is enerzijds nodig om meer gerichte contextinformatie te kunnen aanleveren aan het model voor het genereren van het antwoord. Anderzijds kunnen we op die manier het aantal tokens beperken dat als input gegeven wordt aan het taalmodel, zodat de kosten beperkt worden. Vervolgens gebruiken we een functie om in één beweging de vector index aan te maken in een vector store (Chroma) met behulp van een embeddings model (OpenAI).

Uitvoeringsfase: In deze fase gebruiken we een *ConversationalRetrievalChain* voor de orchestratie van de retriever en de generator (LLM). Als parameters geven we in hoofdzaak het taalmodel mee (OpenAI GPT-4) en het retrieval algoritme. We maakten gebruik van de *EnsembleRetriever* voor hybride search: een combinatie van klassieke lexicale search en semantic search.

Door de krachtige abstracties zijn we in staat geweest om snel en met beperkte code een functionerende question answering toepassing op te zetten. De echte uitdaging zit echter in het robuust maken van de toepassing: het verhogen van de accuraatheid van de antwoorden door het optimaliseren van de data ingestion en de retrieval zodat de meest relevante context gevonden wordt als input voor het taalmodel.

Gebruiksvoorwaarden & Budget

LangChain is open source en gratis te gebruiken onder de MIT licentie. De brondcode is beschikbaar op GitHub.