⛓️

LangChain — Chat with your data

course link: https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data/

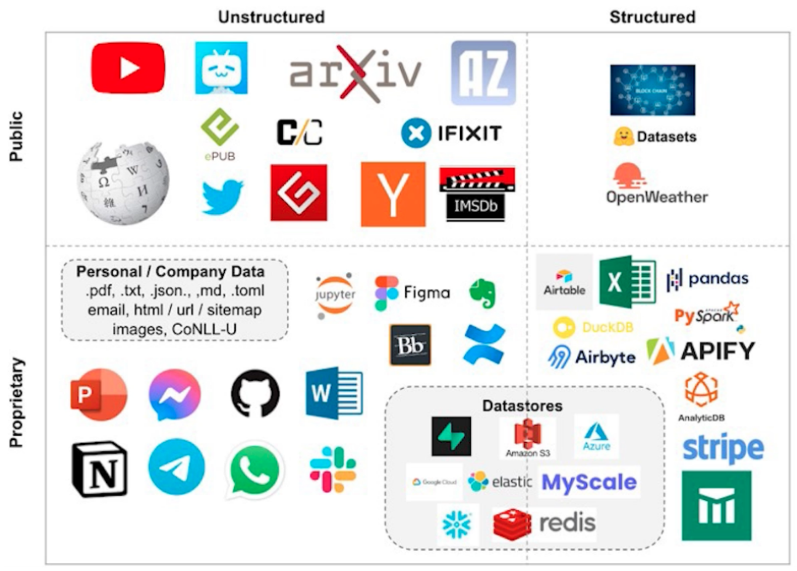
Introduction
Document loading
Document splitting
Vector stores and embeddings
Retrieval
Question answering

Introduction

Welcome 😊

Document loading

- document loaders



  - PDF
    - when loading a PDF, it will be split into pages
    - each page contains text, and metadata about the document it is a part of
  - youtube video transcript
    - OpenAI Whisper
  - URLs
    - WebBaseLoad(f"{URL}"
  - Notion

        - Notion directory loaders

Document splitting

- you need to split documents into smaller chunks
    - `chunk_size` & `chunk_overlap`
- types of splitters
    - text_splitters
- the metadata of sources and source chunks are the same
- context-aware splitting
    - you can split .md files based on headers and sub-headers

Vector stores and embeddings

- use of embeddings → vectors
- ChromaDB is in-memory and lightweight ⇒ perfect for prototyping
- edge cases and failures
    - duplicate documents
    - “third lecture” doesn’t quite work for embeddings ⇒ you get things that aren’t related to this one specifically

Retrieval

- query → retrieve more relevant chunk
- maximum marginal relevance (MMR)
    - sometimes we want to include information that is not the most similar to the query (eg toxicity of mushrooms)
    - get top fetch_K most relevant responses, then choose K responses that should be the most diverse
- self-query (LLM aided retrieval)
- compression: reduce size of output
- we can retrieve without a vector database (TF-IDF, SVM) ⇒ tfidfretriever, svmretriever)

Question answering

- additional methods
    - map_reduce
    - refine
    - map_rerank