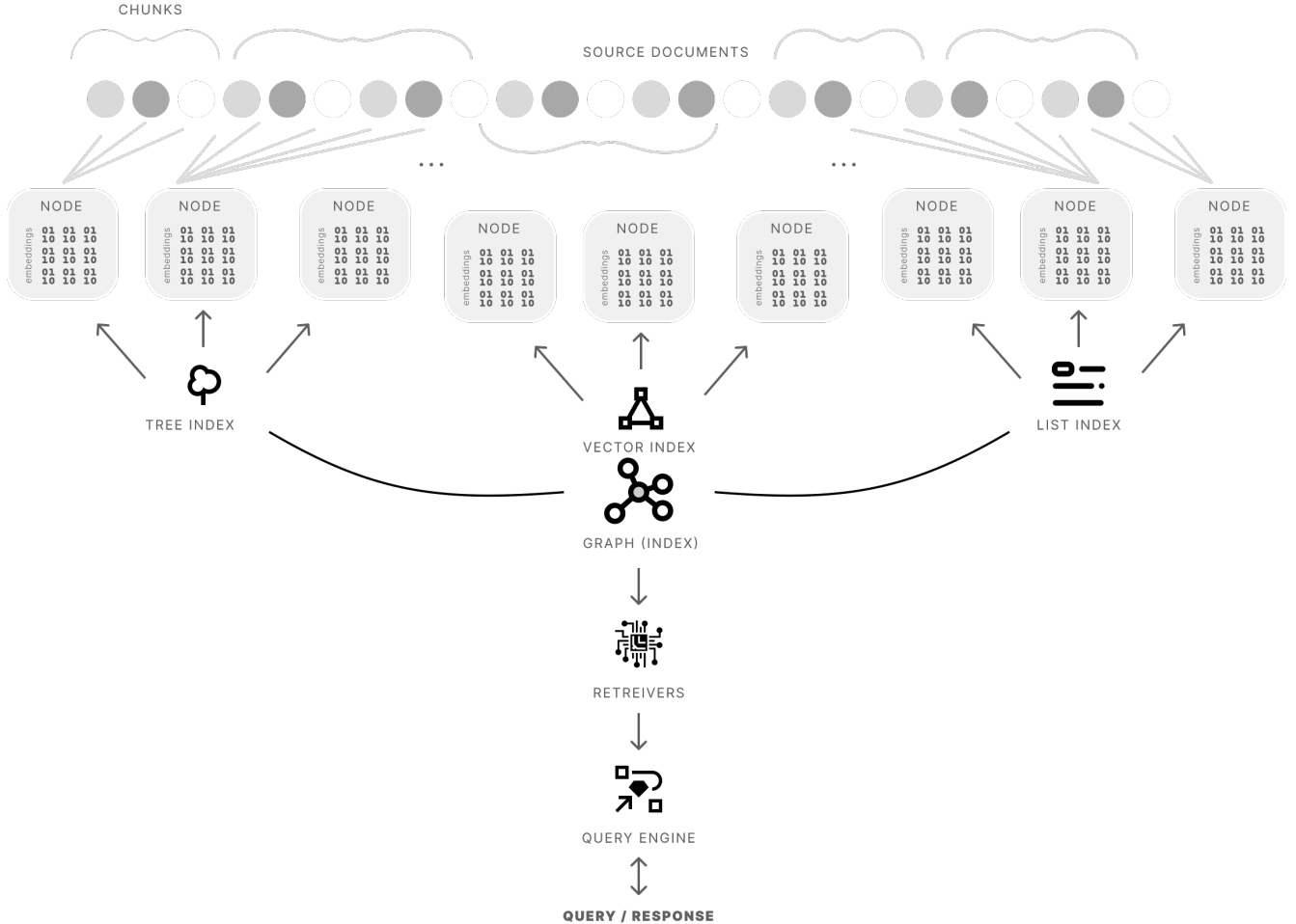


Composable *graphs* with *MMR* queries over *vector* stores.

Composability allows you to define lower-level indices for each document, and higher-order indices over a collection of documents. For *e.g.* imagine defining (1) a tree index for the text within each document, and (2) a list index over each tree index (per document) within your collection.[1]



Querying the index or a graph involves a three main components:

Retreivers →	Response Synthesizer →	Query Engine
A retriever class retrieves a set of Nodes from an index given a query.	This class takes in a set of Nodes and synthesizes an answer given a query.	This class takes in a query and returns a Response object. It can make use of Retrievers and Response Synthesizer modules under the hood.

For the query logic itself we will use maximum marginal relevance or *MMR*. In this we iteratively find documents that are dissimilar to previous results. It has been shown to improve performance for LLM retrievals [2].

The maximum marginal relevance algorithm is as follows:

$$\text{MMR} = \arg \max_{d_i \in D \setminus R} [\lambda \cdot \text{Sim}_1(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in R} \text{Sim}_2(d_i, d_j)]$$

Here, D is the set of all candidate documents, R is the set of already selected documents, q is the query, Sim_1 is the similarity function between a document and the query, and Sim_2 is the similarity function between two documents. d_i and d_j are documents in D and R respectively.

The parameter λ (mmr_threshold) controls the trade-off between relevance (the first term) and diversity (the second term). If mmr_threshold is close to 1, more emphasis is put on relevance, while a mmr_threshold close to 0 puts more emphasis on diversity.

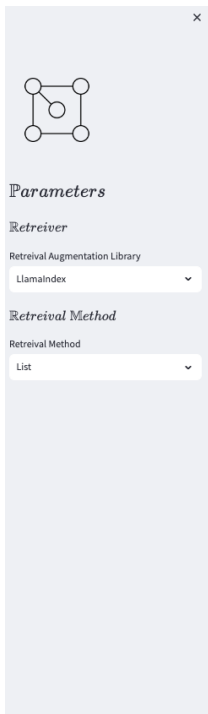
How to use

1. Rename `.env.sample` to `.env` and set environment variables `OPENAI_API_KEY`.

```
# within main folder  
mv .env.example .env  
echo "OPENAI_API_KEY=<your/key/here>" >> .env
```

2. Run App

```
streamlit run main.py
```

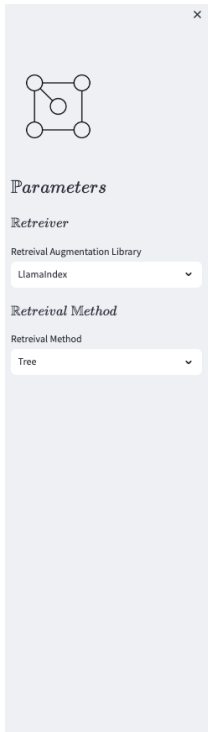


Composable Graphs with Vector indices for LLM queries.

PROMPT

What are the origins of manu?

Submit



Composable Graphs with Vector indices for \mathcal{LLM} queries.

PROMPT

What are the origins of manu?

Submit

The origins of manu are described in the Pankagnividyā, a Vedic text. According to the text, the seed of manu was brought from the wise moon, which is composed of fifteen parts and is the home of our ancestors. The gods then gathered up the seed in an active man, and through an active man, manu was born.

Retrieved Sources

Source (Doc id: d05abb4c-33aa-4567-b41a-1dff43912863): According to the text, the origins of manu are described in the Pankagnividyā, a Vedic text. It s...

Source (Doc id: ffc9d85-1f03-4f65-aed1-57c408ab95bb): verse ma nastoke. The Mantra-Brahmana (II, 1, 8) has the second of these verses only, not the fir...

Source (Doc id: f616eb8e-a620-4471-b559-bdba1cb41c7): Mitra, p. 121.

*246:6 Cf. Taitt. Ar. X, 71 (Anuv. 38, p. 858). Kath. Up. IV, 12-13; above, p. 16...

Source (Doc id: 76364466-3d98-4eac-98af-1f9e97c3a07c): the commentator by ukhshishtadyakramana.

*266:3b This, if rightly translated, would seem to be th...

3. Technologies used

1. openai
2. llamaindex
3. langchain
4. streamlit

Indices

All example indices can be found in the **storage/** folder for loading/reusing in other projects! They are primarily based on CC-commons Indic literature and history corpus from gutenber.