Search Engine with DPR (Nick Bradford)

Motivation

Search engines have two key components: indexing and searching.

At Hebbia, we ingest documents, parse passages (100-token chunks) from them, embed the passages, and store the passage embeddings in the index. At search time, our goal is to surface the best passages, not just the best documents.

Structure

The goal of this project is to give you a taste of what it would be like to build Hebbia from the ground up (sans any of the ML). Phases 1-3 should take about 6 hours.

> 💡 **Learn a lot and impress us.**

Goals

Phase 1 - Document Ingestion and Parsing

- Set up a web server with an input for **ingesting** a file or raw text.
- Determine how to **parse** the input file streams into passages.

Phase 2 - Indexing

- Set up a flow for semantically indexing the parsed passages using a transformer (Dense Passage Retrieval (DPR) is probably the best bet.

    - https://huggingface.co/sentence-transformers/msmarco-MiniLM-L-6-v3 is a good model to use with the `sentence-transformers` library

Phase 3 - Searching

Enable a user to hit your application with a query and return the best passages.

Phase 4 - Cherry on Top (if you have more time)

- Build a frontend to visualize the indexing and search.

or

- Compare semantic search to keyword search

or

- Improve passage parsing logic and/or support other file mime types (e.g. pdf, docx, html)

or

- When you ingest many files at once, indexing cannot happen synchronously in the endpoint. Build out an architecture for async indexing.

or

- Anything else! These are all technical problems we've had to solve so far, but there are so many more on our radar.

We've kept this spec intentionally broad to allow you to wrangle with the same technical issues we've been addressing + to see the new ideas you come up!