Developer Documentation

Decision Records / Generate Embeddings Online

Generate Embeddings Online

Context and Problem Statement

In order to perform a question and answering (Q&A) session over research papers with large language model (LLM), we need to process each file: each file should be converted to string, then this string is split into chunks, and for each chunk an embedding vector should be generated.

Where these embeddings should be generated?

Considered Options

- Local embedding model with `langchain4j`
- OpenAI embedding API

Decision Drivers

- Embedding generation should be fast
- Embeddings should have good performance (performance mean they “catch the semantics” good, see also MTEB)
- Generating embeddings should be cheap
- Embeddings should not be of a big size
- Embedding models and library to generate embeddings shouldn’t be big in distribution binary.

Decision Outcome

Chosen option: “OpenAI embedding API”, because the distribution size of JabRef will be nearly unaffected. Also, it’s fast and has a better performance, in comparison to available in `langchain4j`’s model `all-MiniLM-L6-v2`.

Pros and Cons of the Options

Local embedding model with `langchain4j`

- Good, because works locally, privacy saved, no Internet connection is required
- Good, because user doesn’t pay for anything

- Neutral, because how fast embedding generation is depends on chosen model. It may be small and fast, or big and time-consuming
- Neutral, because local embedding models may have less performance than OpenAI’s (for example). *Actually, most embedding models suitable for use in JabRef are about ~50% performant)
- Bad, because embedding generation takes computer resources
- Bad, because the only framework to run embedding models in Java is ONNX, and it’s very heavy in distribution binary

OpenAI embedding API

- Good, because we delegate the task of generating embeddings to an online service, so the user’s computer is free to do some other job
- Good, because OpenAI models have typically have better performance
- Good, because JabRef distribution size will practically be unaffected
- Bad, because user should agree to send data to a third-party service, Internet connection is required
- Bad, because user pay for embedding generation (see also OpenAI embedding models pricing)