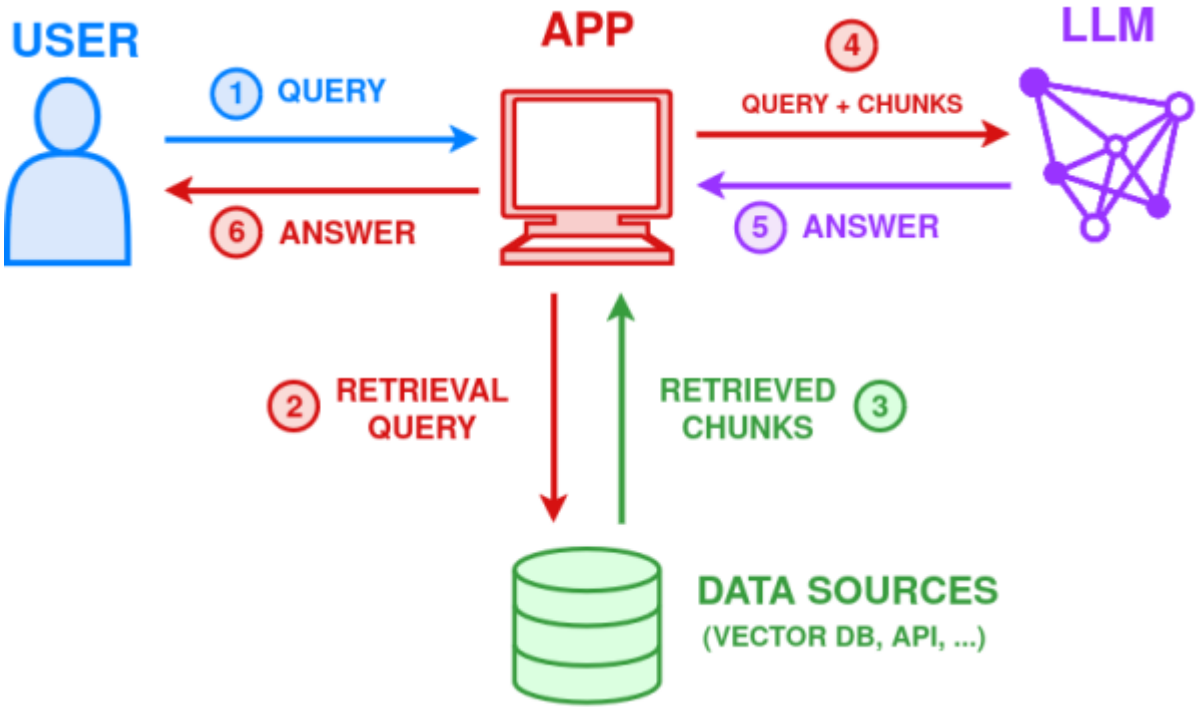
LU12a - RAG Chatbot

RAG (Retrieval-Augmented Generation) ist eine gängige Variante, um Fragen zu benutzerspezifischen Dokumenten zu beantworten.



©
https://www.ridgerun.ai/post/how-to-evaluate-retrieval-augmented-generation-rag-systems

Vorneweg muss man die gewünschten Daten (z.B. Lernunterlagen) ...

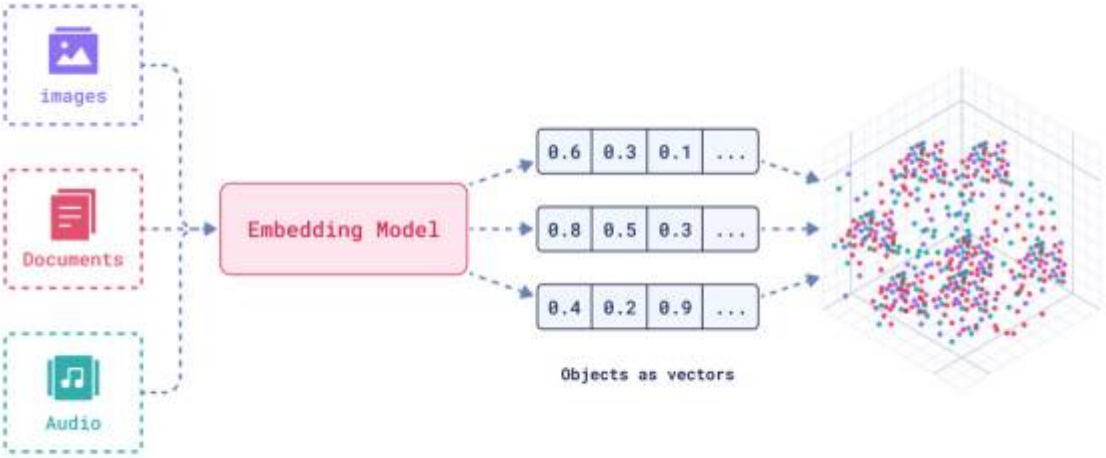
1. ... in Chunks unterteilen
2. ... „embedden“
3. ... in die Vektordatenbank speichern

Der Ablauf funktioniert grob so:

1. Der Benutzer gibt eine Frage ein, welche an die Applikation geschickt wird.
2. Die Frage wird „Embedded“ und an die Vektordatenbank geschickt
3. Von der Vektordatenbank werden die n passendsten Chunks im Klartext an die Applikation zurückgegeben.
4. Die Applikation schickt die originale Frage im Klartext mitsamt den Chunks an ein LLM-Model
5. Das LLM-Model schickt eine Antwort zurück an die Applikation
6. Die Applikation kann z.B. Quellen o. Ä. bei Bedarf ergänzen und die Antwort an den Benutzer zurückschicken.

Vector-Embedding

Bei einem Vector-Embedding werden Daten (oftmals Textblöcke) in Vektoren mit hunderten von Dimensionen umgewandelt.



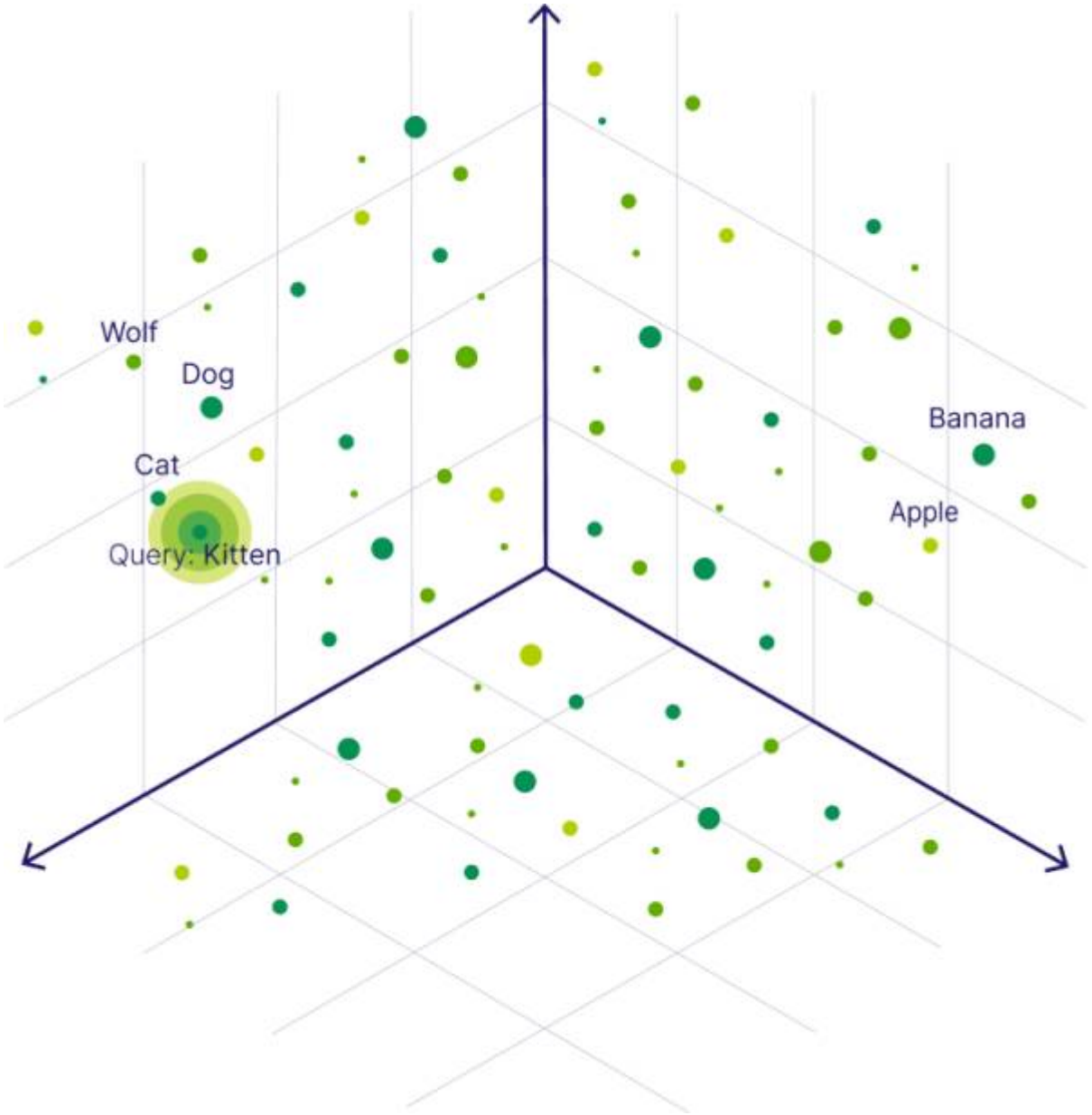
©

https://qdrant.tech/articles/what-are-embeddings/

Vektordatenbank

In einer Vektordatenbank sind Objekte mit ähnlicher Bedeutung aufgrund des zuvor angewendeten Embeddings nahe beieinander. Im nachfolgenden Beispiel sieht man die Tiere links, während die Früchte rechts sind.

Nebst den Zielobjekten können auch Fragen embedded werden. Wird zum Beispiel nach „Kitten“ gesucht, kann die Vektordatenbank mittels einer „Similarity search“ die nächstgelegenen Objekte ermitteln und zurückgeben. In diesem Beispiel also „Cat“.



©
https://nlpcloud.com/de/fine-tuning-semantic-search-model-with-sentence-transformers-for-rag-application.html