RAG

Retriever Augmented Generator

Das Hauptproblem bei der Nutzung von AI-Modellen (wie ChatGPT) in Unternehmen ist, dass diese über extrem fundiertes Wissen verfügen, welches jedoch keine internen Firmendaten beinhaltet. Gerade diese Daten sind jedoch interessante Kandidaten, zu denen man gerne Fragen stellen möchte. Es hat sich gezeigt, dass, wenn man dem AI-Modell neben der Frage auch die relevanten Firmendaten zur Verfügung stellt, das Modell in der Lage ist, Antworten zu finden, die auf den Firmendaten basieren und zugleich das interne Wissen des Modells einbeziehen. Die RAG-Architektur extrahiert die relevanten Firmendaten und liefert sie mit der Frage ans AI-Modellen welches eine spezifische Antwort auf der Basis der Firmendaten generiert.

Definitionen

Retriever bezieht sich auf den Teil des Systems, der relevante Informationen findet.
Augmented im Kontext von RAG bedeutet, dass der Generator durch den Retriever erweitert wird.
Generator ist der Teil, der die eigentliche Antwort erstellt, also zum Beispiel ChatGPT.
Large Language Model (LLM): Die Fähigkeit dieser Modelle liegt in der Bewältigung von NLP-Aufgaben wie Textgenerierung, Übersetzung, Zusammenfassung und Fragenbeantwortung.
Natural Language Processing (NLP) bezeichnet die Fähigkeit von Computern, die menschliche Sprache zu verstehen, zu interpretieren und zu generieren.
Foundation Model: Im Wesentlichen sind dies LLMs, die an riesigen Datensätzen trainiert wurden. Ihre Möglichkeiten reichen über NLP hinaus und umfassen auch Fähigkeiten in Bereichen wie Vision und Audio.
ChatGPT: 'GPT' steht für die Architektur des Modells (Generative Pre-trained Transformer). Dieses Modell besitzt sowohl LLM als auch Foundation-Model-Fähigkeiten.

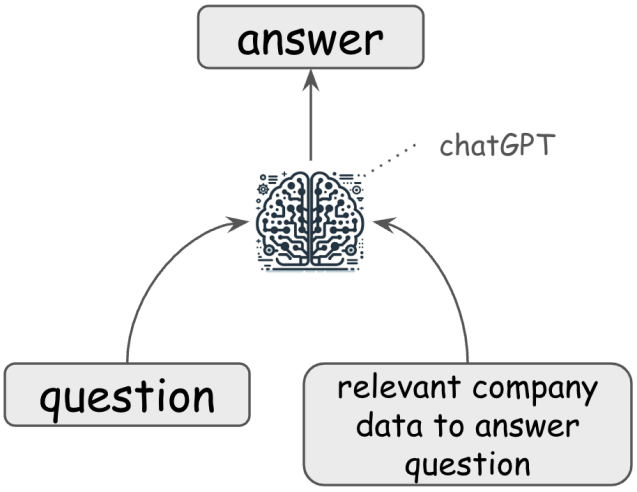

Fig 1: Schema RAG Konzept

Architektur

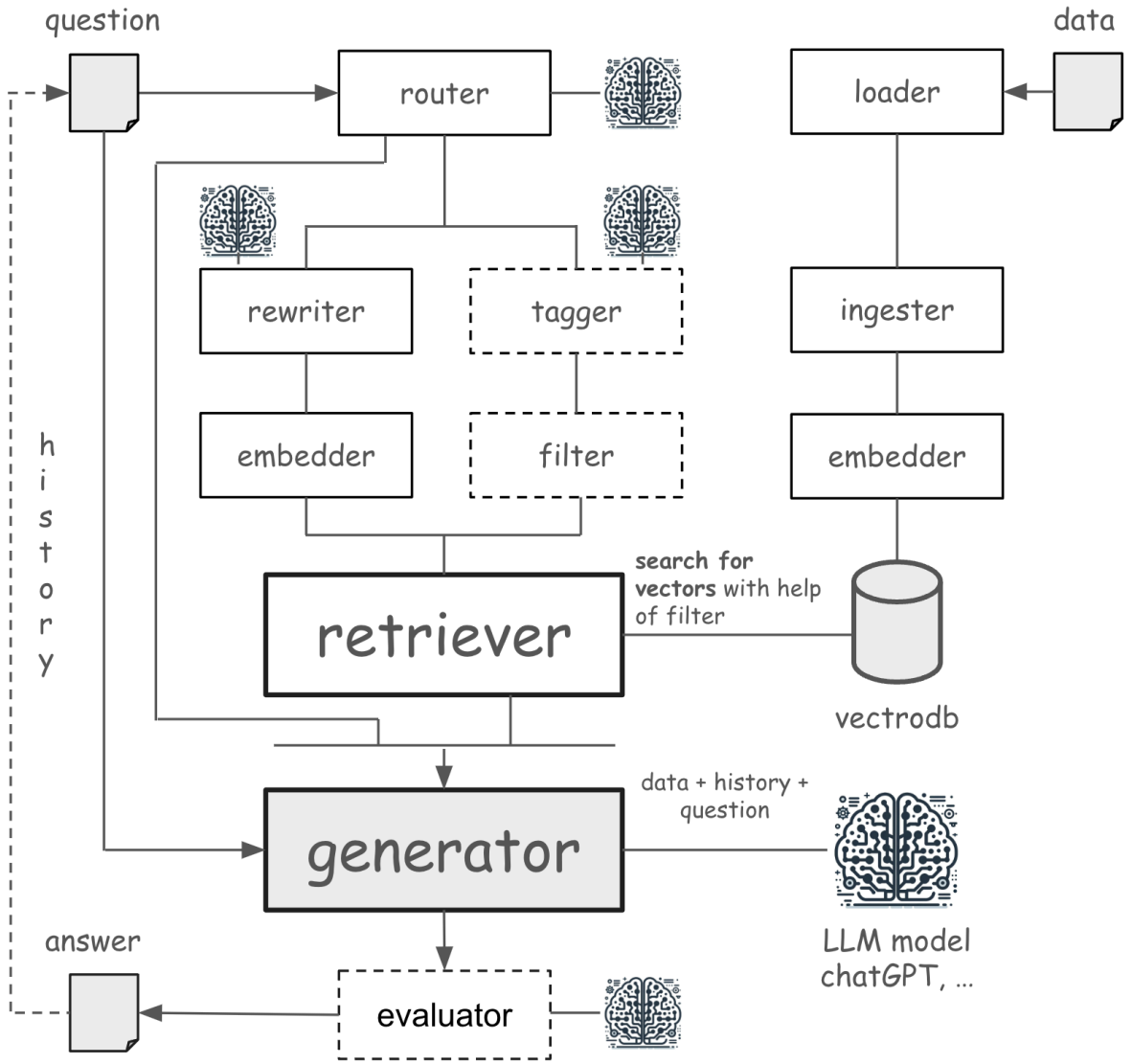

Fig 2: Implementierung RAG

Flow question -> answer	**Flow data**
router: Entscheider ob es den Retriever braucht **tagger**: Extrakter von Metadaten für den Filter **filter:** Schränkt den Datenbereich ein **rewriter:** Optimiert die Frage für den Retriever **embedder**: Transformiert Text in Vektoren **retriever:** Sucht die relevanten Firmendaten **generator:** Generiert die Antwort der relevanten Firmendaten und der Frage **evaluator:** Überprüft die Antwort mit der gegebenen Frage nochmals mit Hilfe des Models	**loader:** Lädt die Firmendaten **ingester:** Bearbeitet die Daten indem er sie z.B. in kleiner gleichmässige Stücke aufgeteilt und in der vectrodb speichert **embedder:** Transformiert Text in Vektoren **history:** Die Fragen und Antworten können als 'History' dem Model immer wieder als relevante Firmendaten mitgegeben werden, damit das Model die Konversation verstehen kann.