Interactive Knowledge Repository powered by LLM

March 04, 2024

CONTENTS

Introduction

Foundation Large language models (LLMs) like GPT, Gemini, etc. can generate human-like text and engage in dialogue, making them well-suited for knowledge management applications. These language models are multilingual, meaning they can be asked to return the answer in a language that can be different from the user's input query language. An interactive knowledge repository powered by an LLM can provide a natural language interface for organizing, retrieving, and contributing knowledge.

This white paper examines the potential of LLMs to enable more intuitive knowledge management through conversational interactions. It provides an overview of LLMs, their current abilities, and how they can be applied to build an intelligent knowledge base. Key topics covered include knowledge representation, techniques for knowledge extraction and synthesis, a proposed architecture for an interactive knowledge repository, and evaluation metrics to assess performance. This paper also offers insights into how LLMs are poised to transform knowledge management through more natural human-computer collaboration.

Problem Statement

The exponential growth of information makes knowledge management an increasingly difficult challenge for organizations. Important knowledge is often scattered across documents, email threads, chat logs, and individual experiences. While knowledge repositories like corporate wikis aim to consolidate information, they lack intelligence and the ability to provide natural language access. Large language models (LLMs) present an opportunity to enable more intuitive knowledge management through conversational interfaces. The core problem is developing an interactive knowledge repository that takes advantage of LLM capabilities in language understanding, natural text fluency, reasoning, summarization, and synthesis to provide a natural language interface for retrieving organizational knowledge. Users could utilize natural language queries and dialogue with the system to search for and contribute knowledge.

Building Blocks

Vector database

Elasticsearch is a popular open-source search and analytics engine that comes out of the box with our analytics product, **tcgmcube**. It allows performing fast searches on data indexed in a distributed, scalable way. Elasticsearch stores data as documents, which can contain nested data structures like vectors. It supports vector similarity searches using cosine distance metrics out of the box. This makes it optimized to store and query dense vector embeddings generated by language models.

Framework for developing applications powered by LLMs

LangChain serves as a framework for crafting language-model-driven applications. This framework facilitates the development of context-aware applications by linking language models to contextual sources. Additionally, it empowers applications with the capability to engage in reasoning tasks utilizing language models, determining responses based on the provided context, and determining appropriate actions.

The LangChain chains and agents are highly customizable, providing a plethora of opportunities for developers to build their own exciting LLM applications. Additionally, it features integrations with a variety of vector databases, language models, Embedding models, document loaders, and other essential components, making it well-suited for industry-specific applications.

Embedding model

Embedding models are used to represent the natural language text in numerical form. The numerical representation can then be used to measure the relatedness between two pieces of text. Our solution supports both the hosted embedding models such as *Text-embedding-3-small* from OpenAI as well as open-source embedding models such as *SFR-embedding-mixtral*, which can be used inside a private cloud without sending the data outside.

Language model

Foundation LLM is used at the end to convert the document chunks into natural language. It adds GenAI capabilities to the RAG solution, such as synthesizing and condensing information, maintaining context, etc. Our solution can leverage hosted LLMs such as GPT, Gemini, etc. as well as open-source LLMs such as Mixtral to enhance data security.

Solution Approach and Methodology

Building a custom interactive knowledge repository involves the following steps:

Storing knowledge in a vector database

The first step is to store the knowledge repository content in a vector database for efficient semantic retrieval. Specifically, the text content is fed as input to the language model in chunks, such as sentence-level or paragraph-level chunks, but before that, we need to extract the text that may be present in the form of different formats or types of files. We provide support for several document types, such as PDFs, HTML, JSON, CSV, and many more. Then it can be converted to a numerical vector representation using an embedding model such as text-embedding-ada-002 or text-embedding-3-small. These numerical vectors are indexed in a vector database, along with some metadata to map back to the original content. Storing the content as vectors allows rapid retrieval based on vector similarity. The database can be searched to find vectors close to a given query vector based on some distance metric, such as cosine similarity. This enables semantic search through the knowledge repository, finding relevant content for natural language queries.

The user asks questions from the knowledge repository

In this step, the user interacts with the conversational interface of the knowledge repository system by asking a question in natural language. For example, the interface could be a chatbot. The goal is to allow the user to query the knowledge repository in a natural conversational manner, as if asking a colleague or search engine. There are no constraints or required formats for the questions. The system needs to be able to handle any type of natural language question related to the knowledge domain. The next steps will focus on processing this natural language user query to retrieve the relevant knowledge.

Embedding model

In this step, the full text of the user's question in natural language is passed as input to an Embedding model, which generates a numeric vector encoding the semantic meaning of the question.

Key aspects that influence the encoding include:

- **Word order and context:** The Embedding model looks at the context of each word in connection to the surrounding words.
- **Syntactic structure:** The model encodes the grammatical structure of the question.
- **Semantic meaning:** Relations between words and concepts are analyzed to capture meaning.

The output is a dense numerical vector representing the full semantic content of the user's question text. This question vector is optimized for similarity comparisons with other encoded text vectors inside a vector database.

Retrieve similar vectors

In this step, the vector representation of the user's question that was generated by the Embedding model is used to retrieve relevant knowledge from the vector database, in this case, ElasticSearch.

Elasticsearch retrieves the top vector matches for the question vector from the database index, enabling the most relevant knowledge content to be surfaced. The similar vectors indicate semantic connections between the user's question and the available knowledge.

Semantic connection refers to how close two vector representations are in the conceptual meaning space. Strong semantic connections retrieve content that is highly conceptually related to the user's information needs, even if the wording differs.

Generate response

In this step, the relevant knowledge content retrieved in step 4 is used to generate a natural language response to the user's question. The vectors most similar to the user's question vector point to the key pieces of content from the knowledge repository that can answer the query. This content is passed to a language model, in this case, GPT-4 Turbo. The language model analyzes the content to produce a response in natural language.

Key aspects that LLM handles in a generation:

- Identifying key facts and concepts from the retrieved content
- Synthesizing and condensing information
- Maintaining context for the user's question
- Producing a readable, conversational response

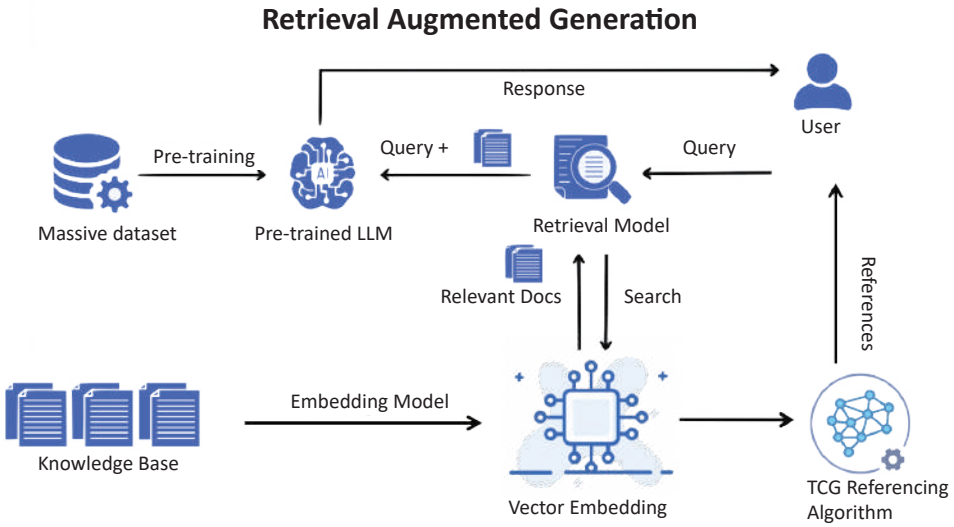
The result is a natural language response tailored to provide the user with information relevant to their query in a consumable way.

LangChain provides us with different chain options that can be used to determine how to pass the retrieved pieces of content to the language model, for example, the StuffDocuments chain, MapReduceDocuments chain, and MapReRankDocuments chain.

Show response

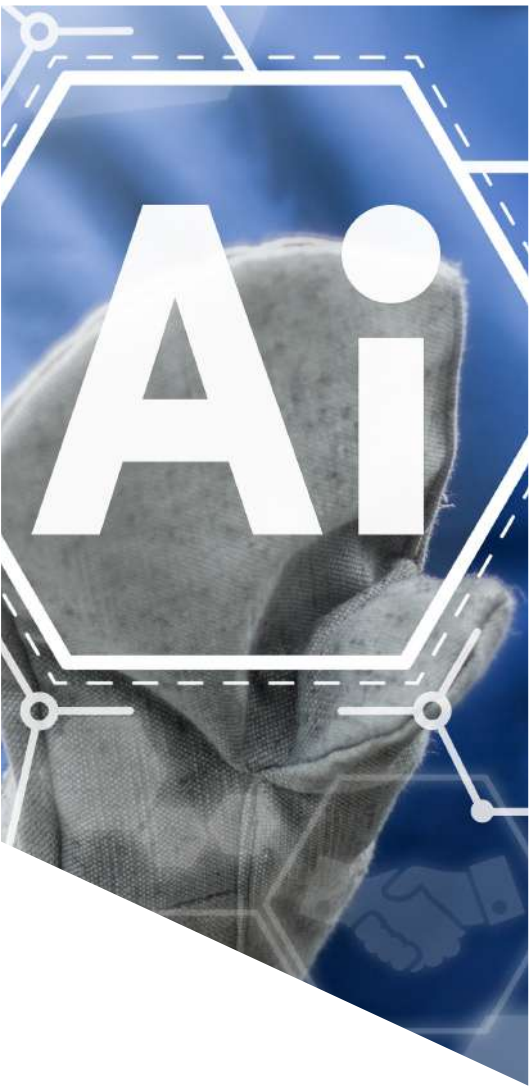
In this final step, the conversational interface presents to the user the natural language response generated by LLM. The system takes the text response and displays it to the user who originally posed the question. This completes the query loop that started when the user asked their question. The goal is to deliver the response to the user in a natural and conversable way. The system aims to emulate how a human expert would provide information relevant to the user's information needs. Showing the final response allows the user to see the answer produced by the knowledge repository system. The user can then provide feedback, ask follow-up questions, or refine their query, enabling an iterative conversational experience.

Architecture Diagram



Conclusion

Large language models like GPT-4 Turbo present new opportunities to transform knowledge management through conversational AI. This whitepaper provides a guide on building an intelligent and natural language-driven knowledge repository using LLM capabilities. The proposed solution demonstrates how LLMs can enhance knowledge discovery through semantic search and synthesize responses from retrieved contexts. Key components include vector databases for knowledge encoding, Embedding models to transform natural language into vector embeddings, LLMs for processing, and conversational interfaces. The customer support chatbot case study exemplifies a real-world application, while the step-by-step methodology outlines development processes. As LLMs continue to advance, they will enable more intuitive human-computer collaboration in gathering, sharing, and utilizing knowledge. Interactive knowledge repositories powered by LLMs can make organizational knowledge more accessible, assist human experts, and unlock new possibilities for knowledge work.



Author

Ashutosh Ranjan
Senior Manager, TCG Digital

TCG Digital is the flagship data science and technology solutions company of 'The Chatterjee Group' (TCG), a multi-billion dollar conglomerate. We leverage hyper-contemporary technologies and deep domain expertise to engage enterprises with full-spectrum digital transformation initiatives in operational support systems, enterprise mobility, app development and testing, cloud and microservices, automation, security, big data, AI/ML, and advanced analytics.

In addition to our digital transformation practices, by using our end-to-end AI and advanced analytics platform, **tcgmcube**, enterprises are extracting highly actionable insights from their invaluable data assets, and achieving Velocity to Value. **tcgmcube** democratizes data science with scalability, performance, and flexibility. For more information, please visit our website at www.tcgdigital.com