prosum
AI-DRIVEN PROFIT OPTIMIZATION

AI KNOWLEDGE RETRIEVAL CHATBOT

Introduction

The Prosum's Knowledge retrieval engine is a cutting-edge solution revolutionizing the way organizations access and utilize vast repositories of information. Built upon state-of-the-art Azure technologies, it represents a breakthrough in information retrieval, seamlessly blending advanced language processing with efficient data mining capabilities.

By harnessing the power of machine learning, our Knowledge Retrieval LLM offers unparalleled accuracy and speed in retrieving, summarizing, and contextualizing information from diverse sources. Whether its uncovering insights buried within extensive documents, providing instant answers to complex queries, or delivering personalized recommendations from company documentation, this Knowledge retrieval engine empowers users to navigate the wealth of knowledge at their fingertips with ease and efficiency.

The Knowledge retrieval engine seamlessly integrates with a multitude of endpoints, including Microsoft Teams, custom websites, mobile applications, CRM systems, and more. Its unique coding framework allows for easy customization to cater for specific client requirements.

Critical Challenges

The design of the knowledge retrieval engine aims to tackle numerous critical challenges across diverse domains, with the challenges encompassing the following:

- **Information retrieval and summarization:** Efficiently retrieve relevant internal or external info from vast amounts of textual data and summarize it concisely, enabling quick access to essential information.
- **Task Automation:** Automation of repetitive tasks such as answering customer inquiries, generating reports, or drafting emails, freeing up human resources for more complex and strategic tasks.
- **Knowledge Discovery:** Uncover insights and patterns within large datasets, aiding businesses, and end users in making data-driven decisions and discoveries.
- **Language Translation:** Support multilingual translation tasks, enabling seamless communication across different languages and cultures.

The Solution Approach

The knowledge retrieval engine integrates both internal and external data sources within the Retriever Augmented Generation (RAG) framework to ground the Large Language Model (LLM) and enhance overall performance. Its primary goal is to facilitate swift and efficient retrieval of requested information for internal staff or a company's external client base, addressing challenges of time-consuming and inefficient data sifting. By implementing this solution, companies can redefine how they extract value from their data in a competitive and fast-paced environment.

Constructed using the Azure technology stack's REST APIs, the knowledge retrieval engine incorporates components like Azure Document Intelligence for extracting information from various document formats and Azure AI Search for optimizing

document indexing and search using vector embedding and semantic search techniques. Azure AI Studio's LLM modelling capabilities are leveraged for configuring, training, fine-tuning, and deploying the chat models. Additionally, Azure Web Applications are deployed to present the chatbot solution through a web interface, serving as the interaction layer between the client and the chatbot, with customization managed on a per-client basis.

A successful implementation hinges not only on the adoption of individual technologies or frameworks like prompt chaining or RAG but on a balanced, multi-pronged approach integrating all these elements to deliver a comprehensive solution. Throughout the implementation process, a test and learn framework is employed to optimize the chatbot according to the clients' specific needs and requirements.

Product Features

Prosum's knowledge retrieval engine offers a roadmap for deploying an LLM model that yields tangible results and enhances efficiency within corporate settings. Key product features include:

- **Data and workflow orchestration:**

  This involves managing the flow of data, tasks, and processes essential for model development, training, and deployment. Organizing and preparing datasets for grounding LLMs from disparate data sources and various formats (pdf, pptx, word, excel, json, txt etc.), ensuring data quality, relevance, and diversity to enhance model performance. The management of workflow design, pipeline automation, version control, resource allocation and error handling.

- **Prompt and Parameter Optimization:**

  Utilizing exclusive coding methodologies alongside cosine similarity algorithms to identify optimal parameter configurations and prompt structures, thereby maximizing the performance of the Large Language Models (LLMs). Including prompt engineering methodologies like few-shot learning, chain of thought, keyword injection, as well as primary, supporting, and grounding content integration, among others.

- **Model Customization:**

  The process of tailoring the model's behaviour, capabilities, and outputs to suit specific tasks, domains, or user requirements. Adjusting model parameters, such as learning rates or layer sizes, through additional training on task-specific data to improve performance on a particular topic or domain. Incorporating feedback loops into the model training framework for continued accuracy improvement and model reinforcement.

- **Model reliability and ethics:**

  Upholding principles of accountability, inclusiveness, reliability, fairness, transparency, privacy, and security in the development, deployment, and use of large language models to ensure their integrity and ethical application.

- **Model serving and integration:**

  Integrating large language models seamlessly into existing platforms and workflows, while ensuring reliable performance, scalability, and accessibility to meet diverse application needs.

Implementation

The usual implementation timeframe spans approximately 3-4 months, subject to the complexity of the client's data sources. Implementation involves configuring the environment, ingesting, processing, and parsing data sources. Following this, tasks include documentation indexing, designing system messages, and optimizing prompts. Subsequently, Parameter and Prompt grid search optimization, model fine-tuning, automation, deployment, and integration take place. The implementation plan further encompasses model monitoring and maintenance services, documentation, as well as regular updates and ecosystem enhancements.