Research

**Enhance Proposal Generation Using Retrieval-Augmented Generation (RAG)**

A hybrid method that merges generative AI with information retrieval, Retrieval-Augmented Generation (RAG) improves the relevance and accuracy of produced material. When a user enters a query into RAG, the system initially checks a database or search index for any pertinent documents or knowledge. A generative model (like GPT) uses this recovered data as input to generate an understandable and useful output. Particularly helpful for activities like content production, question-answering systems, and proposal generation, RAG augments the generative process with real-time retrieval to guarantee that produced content is both factually sound and contextually relevant.

**Workflow of RAG**

A user initiates a query—a request for information, content generation, or some other kind—into a project's Retrieval-Augmented Generation (RAG) workflow. The system then uses a search engine, such as Elasticsearch, to obtain pertinent information from a knowledge base, such as case studies or prior performance statistics. To ensure that the most relevant information is prioritized, the documents that were retrieved are rated according to their relevance to the query. Preprocessing the papers to sanitize the text or remove important portions is an optional extra. The next step is to feed the collected data into a generative AI model (like GPT) so it can create a personalized answer or piece of content that is directly related to the user's query. After that, it goes through quality control post-processing to make sure the output is formatted correctly and follows all the rules. The consumer receives the finished product and has the opportunity to examine it and offer comments. The system can use this feedback to iteratively enhance its retrieval and generation processes, making them better over time. Complex jobs like proposal development, content production, and report writing benefit greatly from RAG because of this technique, which guarantees factually accurate and contextually relevant generated content.

**Example Use case of RAG**

**Retrieval**: A user asks the chatbot, “Generate a proposal for a cloud migration project.” The system queries the past performance database (Elasticsearch) and retrieves:

* A case study for a similar cloud migration project.
* A list of relevant technical solutions implemented in previous projects.
* Client testimonials from past projects.

**Augmented Generation**: The system then:

* Generates a section of the proposal describing the technical approach, using data from past performance.
* Writes a summary of the retrieved case study, tailoring it to the new client's needs.
* Adds relevant testimonials and adapts the language to match the new proposal’s tone and format.

**References**

Text generation from: ChatGpt (open AI)

https://chatgpt.com/c/67992bec-02c0-800e-abe5-1d3f1f2c112d