**Summary**

Below is an overview of a Copilot agent created to write and analyze Rockefeller Asset Management’s (RAM’s) Due Diligence Questionnaire (DDQ) and Request for Proposal (RFP) reports. This documentation will provide an overview of how the agent was built and key factors to consider for both current usage and future improvement. The goal is to show that AI agents have become accessible tools for RAM and to use this document as an introduction to creating agents across future team needs. In the era of AI enhanced productivity and tools, it is important to remain concise and, more importantly, transparent so future users can see disambiguated reasoning and clear production steps.

**Agent Construction**

All the hard work it took to create this agent was fortunately done by years of people that came before me. Over the years, members of RAM’s Sales and adjacent teams have been saving DDQ and RFP reports across time in SharePoint. The foundation of this agent is built entirely on this SharePoint link. The agent is given access to the entire SharePoint, so it pulls beyond only the dedicated RFP folder. A very important characteristic of this folder was the existing time series labeling of all the documents within it. This allows for Copilot to sort documents by time, aligning with the team’s needs. It is important to note here that as a tradeoff to having access to the entire SharePoint, writing the correct prompt becomes more important given there are more avenues the agent could use to search for an answer to the query.

Instructions for the agent were quite basic, and were generated using the “describe” feature in Copilot:

<describe agent>

<knowledge>

The current organizational structure is year-quarter, which can actually be extended to monthly analysis as well given the large amount of accompanying “month-end” or similar files in the SharePoint. While labelling is done only by time, the agent can view changes across strategies by cross examining documents. If in the SharePoint, not additional upload onto Copilot is necessary.

**How Does it Work?**

Copilot’s agent feature operates using the retrieval-augmented generation (RAG) framework. This can be broken down as follows:

Embed → Retrieve (via Azure Cognitive Search) → Inject into Prompt with GPT-4 → Response

* Embedding is the process of converting text into numerical vectors that capture semantic meaning. This step allows the system to compare content based on similarity of meaning rather than just keywords, enabling more intelligent and relevant search results
* The retrieval process uses the embedded query to search across the predefined SharePoint-hosted documents via **Azure Cognitive Search, an internal Copilot semantic mapping**. It identifies and retrieves the most relevant text snippets or document sections related to the query according to the embedding similarity.

A graph with lines and points

AI-generated content may be incorrect.A diagram of a person and queen

AI-generated content may be incorrect.

* The retrieved content is packaged alongside the original user query into a structured prompt, which is then passed into OpenAI’s **GPT-4,** the large language model responsible for reasoning and language generation in this process.
* GPT-4 processes the input and generates a natural-language response that is grounded in the retrieved content. This ensures that answers are not only fluent but also contextually relevant and based on actual organizational knowledge.

**Prompting and Usage Ideas**

Below is a screenshot of some of the sample prompts tested. A general rule of thumb is the more specific, the more accurate the result will be given the earlier point about a wide database.

<sample prompts>

**Future Improvements and Best Guidance**

A recommendation I would make to “future proof” structure is to continue diligently marking documents chronologically, and if smaller time intervals (like weekly or daily) is ever needed to either start noting this within text or to discretize the documents down to monthly (or smaller) intervals further.

Some more brief best practices to follow:

* Never delete data or previous work – many documents which seem unnecessary today may help to improve the future searching mechanism of the agent.
* AI hallucinations – there is occasional incorrect source citing in some of the output; therefore it is very important to audit the results of the queries with empirical data. More beta testing is encouraged on certain edge cases to identify what kinds of questions may cause increased incorrect outputs.