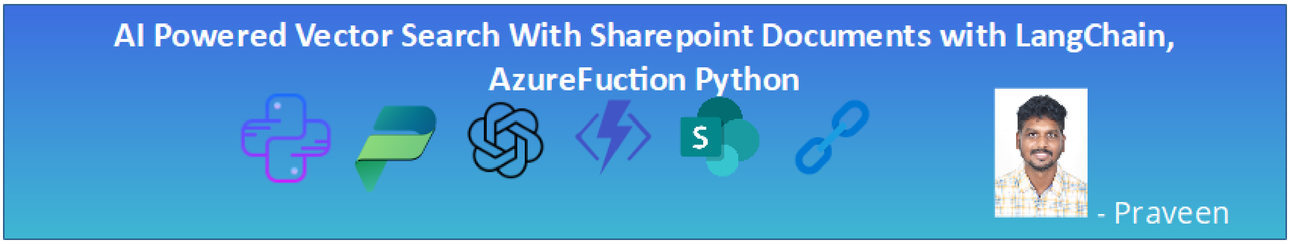
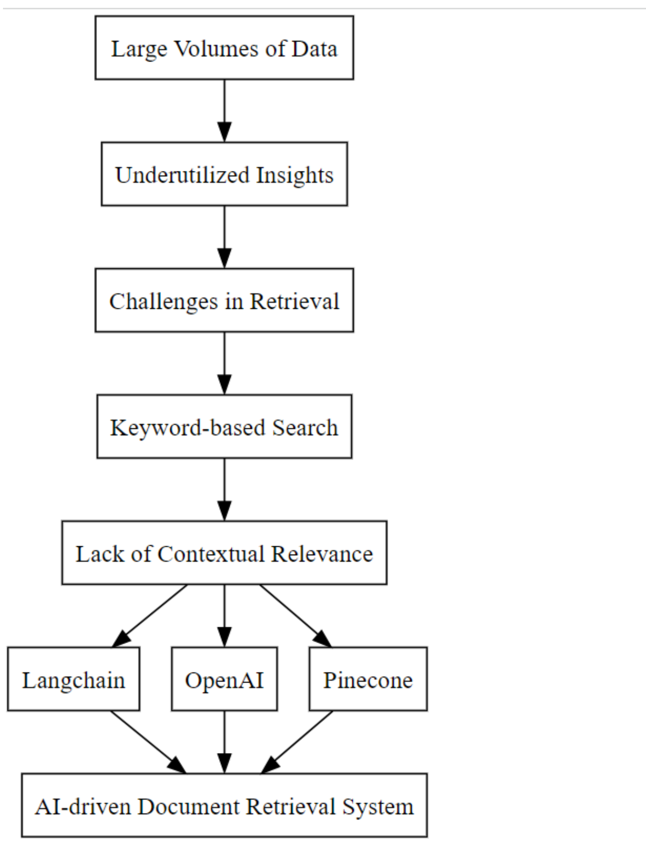


Introduction

In this digital age, the ability to extract insights from unstructured text data is vital. Whether it's analyzing customer reviews, extracting insights from research papers, or understanding the sentiment behind social media posts, text analysis plays a crucial role in many aspects of our lives.

One powerful method for analyzing text data is by using vector embeddings. Vector embeddings are mathematical representations of words or documents that capture their semantic meaning. In this blog post, we'll discuss how to use Azure Functions, SharePoint, and OpenAI to create vector embeddings of documents and store them in Pinecone, a vector database.



What are Vector Embeddings?

Imagine you have a library full of books and you want to categorize them based on their content. One way to do this is to read each book and manually categorize them. But what if you have thousands or even millions of books? This task becomes nearly impossible.

Vector embeddings provide a solution. They transform text into high-dimensional vectors in a way that similar texts are close to each other in this vector space, and dissimilar texts are far apart. These vectors can then be used to perform tasks like semantic search, text classification, and more.

Using Azure Functions for Text Processing

Azure Functions is a serverless solution that allows you to run small pieces of code (functions) without worrying about a whole application or the infrastructure to run it. For our case, we're using an Azure Function to process documents from SharePoint, generate vector embeddings using OpenAI, and store the embeddings in Pinecone.

Fetching Documents from SharePoint

SharePoint is a popular document management and collaboration tool from Microsoft. We use the SharePoint REST API to fetch documents stored in a SharePoint library. The documents can be in various formats like PDF, Word, Excel, or plain text.

Generating Vector Embeddings with OpenAI

Once we have the documents, we need to transform them into vector embeddings. We use OpenAI's text-embedding-ada-002 model for this purpose. OpenAI's text-embedding-ada-002 is a powerful language model that has been trained on a diverse range of internet text.

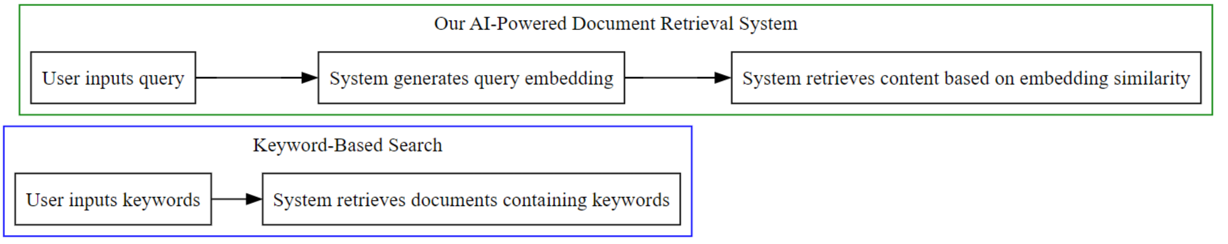
We use this model to generate embeddings for each document. To manage the cost and token limitations of the model, we split each document into smaller chunks and generate embeddings for each chunk.

Storing Embeddings in Pinecone

After generating the embeddings, we store them in Pinecone. Pinecone is a vector database that's optimized for storing and searching high-dimensional vectors.

We create an index in Pinecone and upsert the embeddings into this index. This allows us to later search the index using vector similarity, which can be much more effective than traditional keyword-based search when we want to find similar documents.

Keyword-Based Search vs. Our AI-Powered Document Retrieval System



Traditional Keyword-Based Search

Traditionally, document retrieval systems have relied on keyword-based search. This type of search matches the exact words or phrases that users enter as queries. While keyword-based search can be effective when the user knows the exact terms used in the documents, it has several limitations:

Limited by Exact Matches: Keyword-based search relies on exact matches. If the document uses different words or phrases than those used in the search query, even if they have the same or similar meanings, the document may not be retrieved.

Lack of Context Understanding: Keyword-based search does not understand the context. It treats the query words independently and doesn't understand the relationship between words. As a result, it can miss relevant documents or return irrelevant ones.

Inability to Rank Relevance: While keyword-based search can tell you whether a document contains specific words, it can't tell you how relevant the entire document is to your query. It lacks the ability to rank the documents based on their overall relevance to the query.

Our AI-Powered Document Retrieval System

Our solution overcomes these limitations by leveraging the power of AI, specifically natural language processing (NLP) capabilities of OpenAI's language model. Here's how our solution improves upon traditional keyword-based search:

Semantic Understanding: Instead of relying on exact keyword matches, our AI-powered system understands the meaning of the query. It can find documents that are semantically related to the query, even if they don't contain the exact words. This leads to more relevant results.

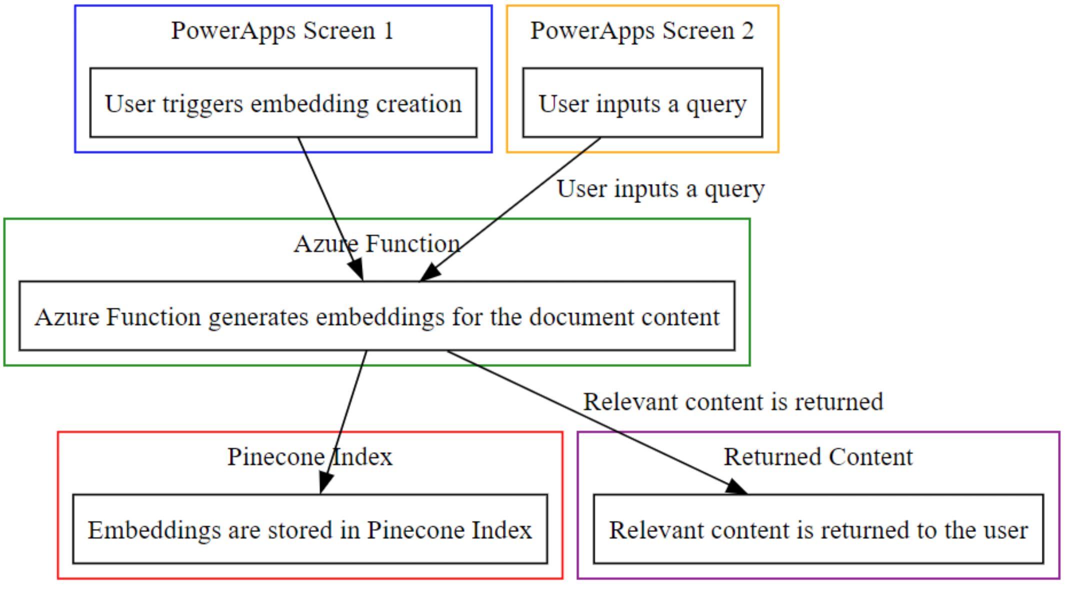
Context-Aware: Our system understands the context of the query. It treats the query as a whole, rather than a collection of independent words. This enables it to better understand the user's intent and return more relevant documents.

Relevance Ranking: Our system doesn't just tell you whether a document is relevant or not; it tells you how relevant each document is. It ranks the documents based on their relevance to the query, helping users find the most useful documents faster.

Sources Referencing: In addition to providing the answer, our solution also provides the sources (documents) that were used to construct the answer. This allows users to trace back the information and explore further if needed.

In conclusion, our AI-powered document retrieval system offers a more efficient, accurate, and context-aware alternative to traditional keyword-based search. It not only helps users find the information they need faster but also provides them with more relevant and complete results.

High-Level Architecture



The high-level architecture of our solution includes four primary components:

1. SharePoint Document Library:

This is where users' documents are stored. SharePoint was chosen because of its widespread use in businesses, and its robust document management features.

2. Azure Function:

This serverless computing service runs on-demand code without needing explicit provision or infrastructure management.

The Azure function we developed is responsible for embedding documents from the SharePoint library using OpenAI's language model, and then storing these embeddings in a Pinecone vector index.

3. Pinecone Vector Index:

Pinecone is a vector database service that's ideal for storing and retrieving embeddings generated by machine learning models. In our case, we use Pinecone to store the document embeddings generated by our Azure function.

4. Custom Connector:

The Custom Connector is used to link our Power App with the Azure Function, allowing the Power App to communicate with the Azure function and thus retrieve the appropriate documents.

5. Power Apps:

Power Apps is a suite of applications, services, connectors, and data platforms that provides a rapid application development environment. In our solution, we developed a Power App to serve as the user interface for our document retrieval system.

Libraries and Frameworks Used

Our implementation uses several Python libraries and frameworks. Some of the most important ones include:

1. Office 365 Python SDK:

We use this SDK to interact with SharePoint and fetch documents from the library.

2. OpenAI's Python API:

This is used to generate embeddings for each document using OpenAI's language model.

3. Pinecone's Python SDK:

We use this SDK to interact with Pinecone's vector database service.

4. Azure Functions for Python:

This library allows us to create an Azure Function with Python that can be triggered through an HTTP request.

5. LangChain Framework:

This is a Python-based framework for building language models chains. We use it to create a retrieval model chain that can return relevant documents based on a query.

Business Use Cases

The potential business use cases for our solution are virtually limitless. Any business that deals with a large number of documents and needs a fast, efficient way to retrieve them can benefit from our solution. This includes, but is not limited to, law firms, medical facilities, corporations, government agencies, educational institutions, and more.

Let's explore some potential use cases for such a system:

Corporate Knowledge Base

Companies have extensive internal documentation detailing their policies, procedures, and guidelines. A conversational AI can help employees navigate this information more efficiently. For example, an employee could ask, "What is our data privacy policy?" or "What's the process for expense reimbursement?" or "How do I request annual leave?" The AI system would then fetch the relevant information from the documents in the library.

Project Management

Project managers could use the AI to keep track of project progress, requirements, and action items. Questions could include, "What are the requirements for the new feature in Project X?" or "What were the action items from the last project meeting?" or "Who is overseeing the QA process in Project Y?"

Human Resources

The AI can assist in answering questions related to HR policies and benefits, such as "What is the company's maternity leave policy?" or "How do I apply for health insurance benefits?" or "What professional development opportunities does the company offer?"

Product Documentation

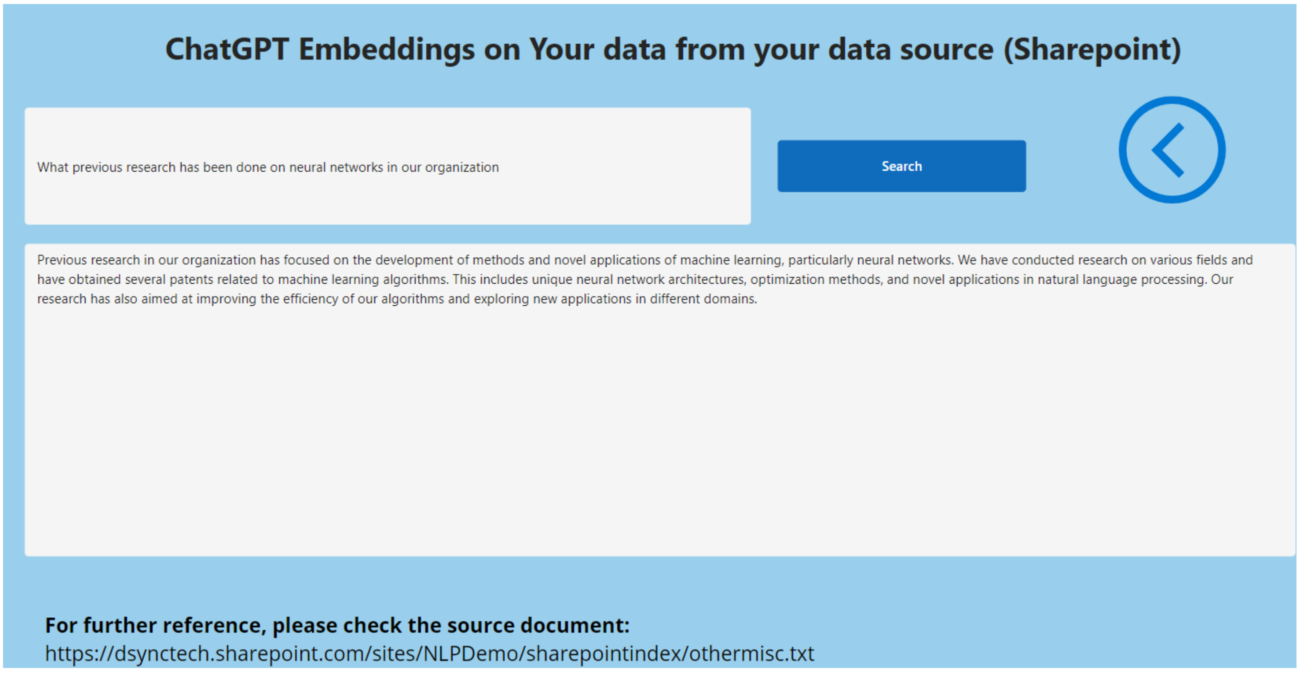
Customers can use the AI to get information from product manuals and documentation, asking questions like "How do I install the software on a Windows machine?" or "What do I do if the software does not start?" or "What are the system requirements for the latest product?"

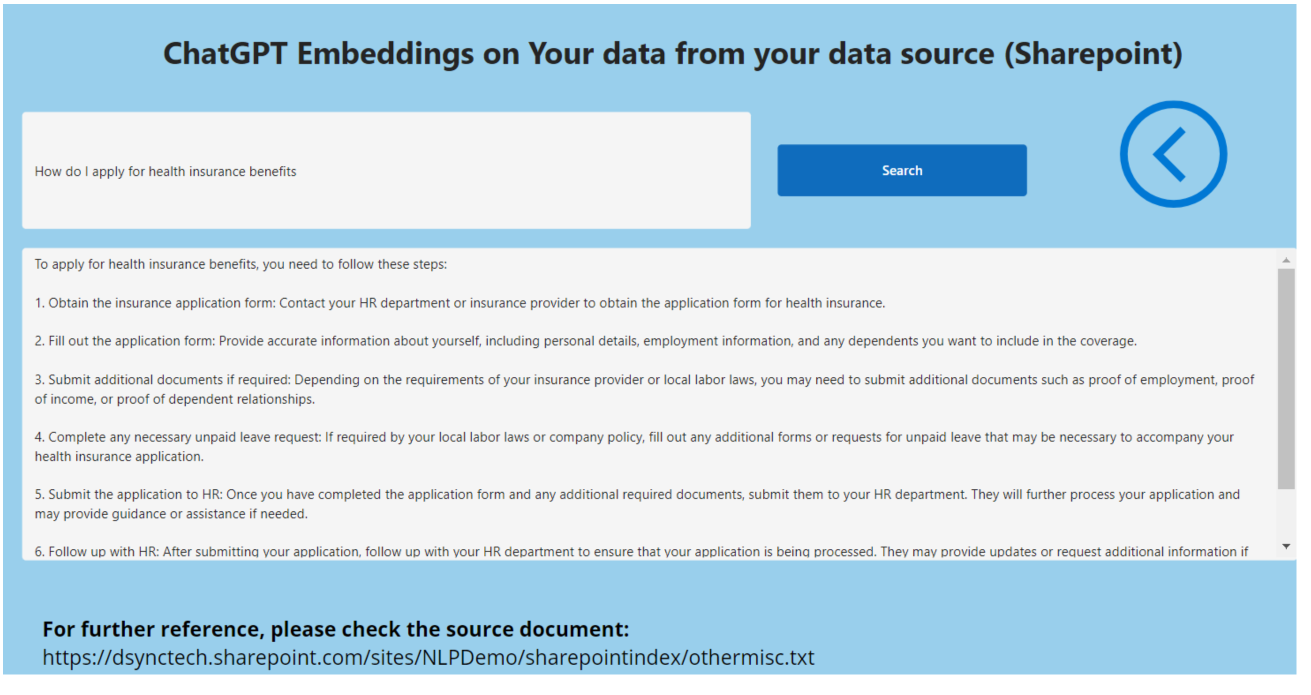
Research and Development

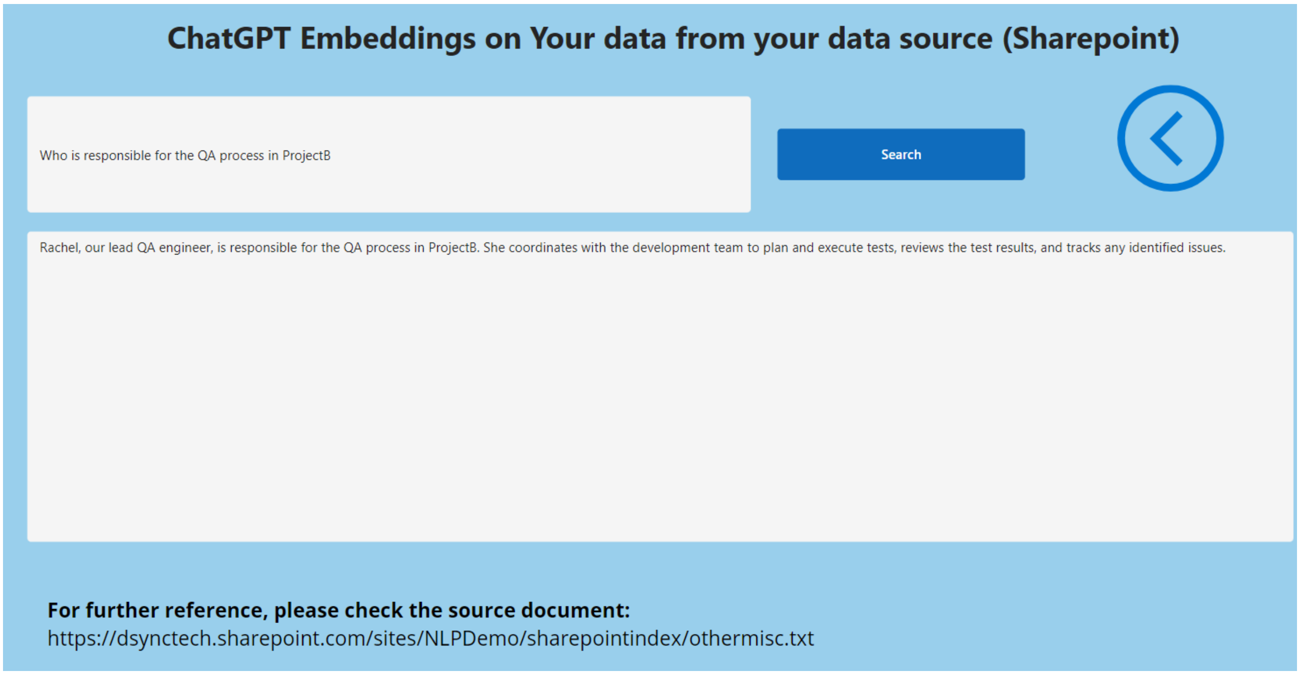
Researchers and developers can use the AI to quickly access previous research, specifications for new products under development, and patent information. They could ask, "What was the focus of our previous research on neural networks?" or "What are the specifications for our new product under development?" or "What patents do we have related to machine learning algorithms?"

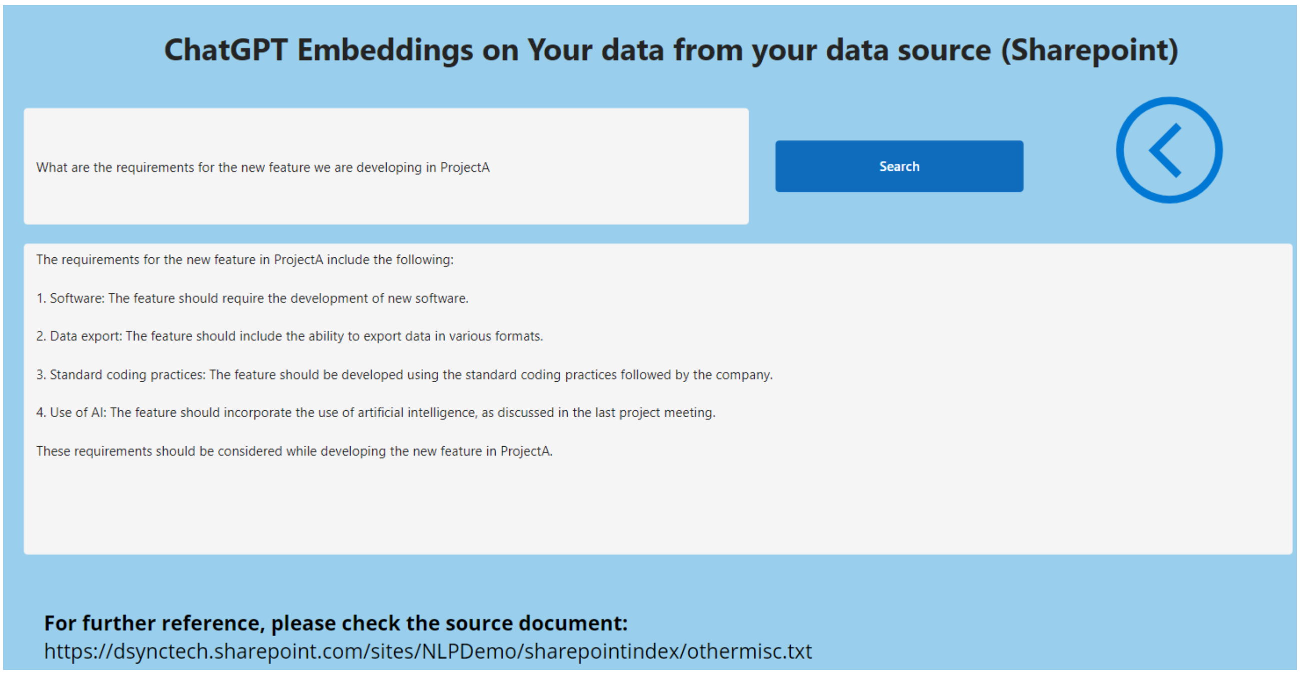
Each of these use cases represents a potential application of a conversational AI system in a corporate environment. By leveraging the power of AI and knowledge graphs, businesses can make their vast knowledge bases more accessible and usable, leading to increased efficiency and productivity.

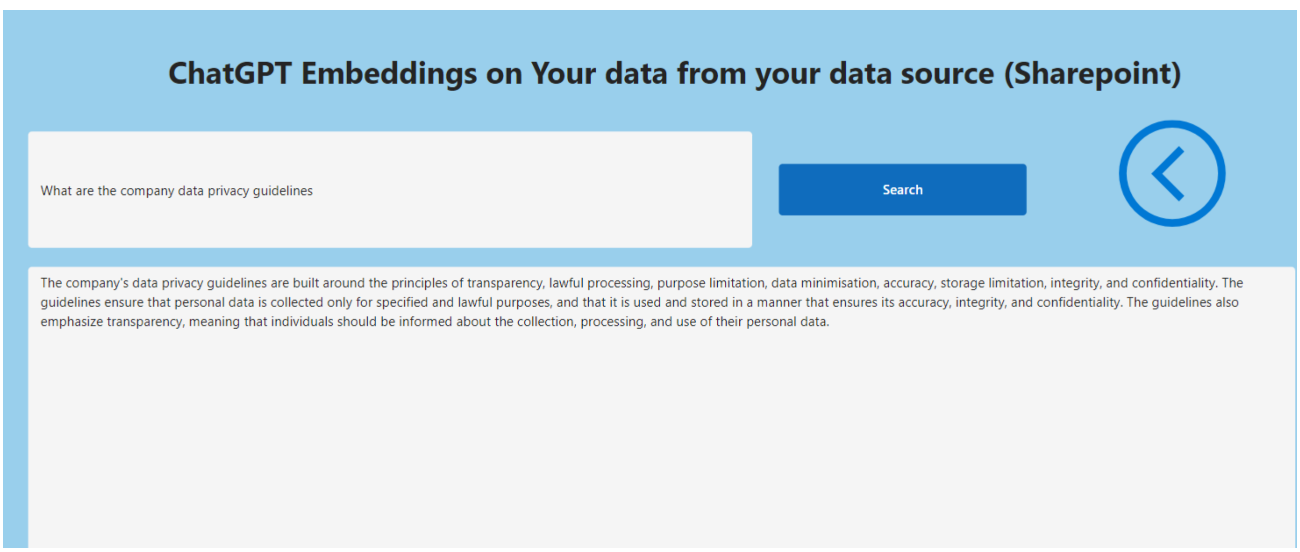
Check Out Our Application's Results in These Screenshots:






ChatGPT Embeddings on Your data from your data source (Sharepoint)
Who is responsible for the QA process in ProjectB
Search
Rachel, our lead QA engineer, is responsible for the QA process in ProjectB. She coordinates with the development team to plan and execute tests, reviews the test results, and tracks any identified issues.
For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/othermisc.txt


ChatGPT Embeddings on Your data from your data source (Sharepoint)
What are the requirements for the new feature we are developing in ProjectA
Search
The requirements for the new feature in ProjectA include the following:
1. Software: The feature should require the development of new software.
2. Data export: The feature should include the ability to export data in various formats.
3. Standard coding practices: The feature should be developed using the standard coding practices followed by the company.
4. Use of AI: The feature should incorporate the use of artificial intelligence, as discussed in the last project meeting.
These requirements should be considered while developing the new feature in ProjectA.
For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/othermisc.txt


ChatGPT Embeddings on Your data from your data source (Sharepoint)
What are the company data privacy guidelines
Search
The company's data privacy guidelines are built around the principles of transparency, lawful processing, purpose limitation, data minimisation, accuracy, storage limitation, integrity, and confidentiality. The guidelines ensure that personal data is collected only for specified and lawful purposes, and that it is used and stored in a manner that ensures its accuracy, integrity, and confidentiality. The guidelines also emphasize transparency, meaning that individuals should be informed about the collection, processing, and use of their personal data.

For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/othermisc.txt

ChatGPT Embeddings on Your data from your data source (Sharepoint)

How does our expense reimbursement process work

Search

The expense reimbursement process starts with the employee submitting an expense report. This report should include all receipts for the expenses incurred. If an expense is above $25, it must be included in the report.

Once the expense report is submitted, it is reviewed by the employee's manager. The manager will assess the expenses and determine if they are eligible for reimbursement according to company policies.

If the manager approves the expense report, it is then sent to the Finance department for processing. The Finance department will review the expenses and ensure they comply with company policies and procedures.

The reimbursement process typically takes approximately two weeks from the time the expense report is approved by the manager. During this time, the Finance department will verify the expenses and prepare the necessary paperwork for reimbursement.

Once the reimbursement is processed, the employee will receive the approved amount through their preferred payment method, such as a direct deposit or a check.

It is important for employees to keep track of their expenses, submit complete documentation, and adhere to company policies to ensure a smooth and timely reimbursement process.

For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/othermisc.txt

ChatGPT Embeddings on Your data from your data source (Sharepoint)

Who succeeded Jane Smith as the CEO of MWP in 2006

Search

Robert Johnson succeeded Jane Smith as the CEO of MWP in 2006.

For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/mwp.txt

ChatGPT Embeddings on Your data from your data source (Sharepoint)

significant event happened to MWP in 1998

Search

In 1998, MWP, a company in the IT services industry, went through a series of corporate re-organizations and had an initial public offering. During this time, the company also changed its name to MWP Technology Solutions. The parent company, MWP, faced challenges but continued to remain a major player in the industry. Additionally, former Infosys President, David Williams, replaced the CEO of MWP. This significant event led to MWP being listed on NASDAQ in the same year.

For further reference, please check the source document:
https://dsynctech.sharepoint.com/sites/NLPDemo/sharepointindex/mwp.txt

Conclusion

Our AI-powered document retrieval solution leverages SharePoint, Azure Functions, OpenAI, Pinecone, and Power Apps to provide a fast, efficient, and accurate way to retrieve documents. By using AI to generate document embeddings and a vector database service to store and retrieve these embeddings, our solution can significantly reduce the time and effort required to find relevant documents. Plus, with our user-friendly Power App interface, users can easily search for and retrieve the documents they need.