Overview
On July 25, 2023, the Azure Cognitive Search team held an AMA on Microsoft Tech Community. The live hour of Q&A provided members the opportunity to ask questions and provide feedback to the product team. We hope you join us live next time!

Resources

- AI + machine learning documentation
- Azure AI overview
- Azure AI Services: Generate tangible value for your organization quickly with AI Services for common business processes. Add cognitive capabilities to apps with APIs and AI services. Includes Azure Bot Service, a comprehensive development environment for designing and building enterprise-grade conversational AI.
- Azure Machine Learning: Enterprise-grade machine learning service for the end-to-end ML lifecycle
- Azure Cognitive Search: Enterprise scale search for app development
- Azure Bot Service: A comprehensive development environment for designing and building enterprise-grade conversational AI
- Azure Databricks : Design AI with Apache Spark-based analytics
- Azure Kinect DK: Build for mixed reality using AI sensors
- OpenAI Service: Apply advanced language models to a variety of use cases

Introduction

Welcome to the Azure Cognitive Search AMA! View the list of introductions in this thread.

General Discussion

Q: Plug In environment roadmap - I've seen technical specs. how 3rd parties will build. Specific to parsing/reading/integrating docs. What is timing for integration/rollout? Reference to tech. articles showing partner examples? (link)

A: Plugins support in Azure OpenAI is currently in a limited private preview. For now, I'd recommend checking out these resources on additional capabilities that you can leverage:
- Function calling is now available in Azure OpenAI Service - Microsoft Community Hub

- Introducing Azure OpenAI Service On Your Data in Public Preview - Microsoft Community Hub

In particular, function calling allows you to do all the same things you could do with plugins. The main difference is that with functions, you're in control of the orchestration and need to call out to the plugins on the client side.

Q: Could you provide insights into the scalability and performance aspects of the Azure OpenAI interface? How does it handle large-scale AI workloads and ensure efficient resource utilization? (link)

A: The biggest thing to keep in mind when it comes to the scalability of Azure OpenAI service are the quota and limits. I'd start by taking a look at this page that talks about the limits. It's also important to learn how to manage the quota for your service.

In terms of performance, the best thing I can recommend is trying out your prompts in Azure OpenAI to understand the performance. One big factor for performance is the length of your prompts, particularly the number of output tokens.

Q: How can I use this if OpenAI is not currently supported in my region? (link)

A: I'd recommend creating an Azure OpenAI service in a region close to your search service. There will be some added network latency but it should be manageable. You can check out the regional availability of different models here: Azure OpenAI Service models - Azure OpenAI | Microsoft Learn

We're also working hard to add new Azure OpenAI regions!

Q: What is the approach recommended to store a chatbot conversation history for future context? (link)

A: I would highly recommend you check out this blog post from Semantic Kernel which discusses memory management, There is also this video on the topic.

Q: OpenAI studio (Chat Playground) is able to connect to data sources only if they are in public network, which not the ideal scenario in corporate environments. Any plans to add private endpoint connectivity to studio? (link)

A: Supporting private endpoints is part of that feature roadmap and ETA is the coming quarter (Q4 '23) as per the team supporting it.

Q: How will vector search change or improve upon the pattern in the blog post here https://techcommunity.microsoft.com/t5/ai-applied-ai-blog/revolutionize-your-enterprise-data-with-chatgpt-next-gen-apps-w/ba-p/3762087 and sample code here https://github.com/Azure-Samples/azure-search-openai-demo? (link)

A: You are absolutely correct that vector search will have a very positive impact on this from the ability to get the most relevant results, and we have already updated the sample to leverage vectors (check out the "embedding" field).

In fact, we believe that it is best to leverage Hybrid Search along with the reranking layer of Semantic Search to get the best results. I talk a bit about Hybrid Search in my blog post.

Q: How does the Azure OpenAI interface integrate with Microsoft Azure's existing services and platforms to enhance AI capabilities? (link)

A: One good example of how Azure OpenAI integrates with other products is vector search. You can Azure OpenAI to create embeddings from your content and load them into Azure Cognitive Search. We also created Azure OpenAI on your data to help you chat over data that lives in your search service: Introducing Azure OpenAI Service On Your Data in Public Preview - Microsoft Community Hub

Azure OpenAI service also supports function calling which gives you another way to integrate Azure OpenAI with other Azure services.

Q: I'm applying for a job where the request is for Tech to do more to help the business, lean into business area, extract needs and deliver on Business needs and I'm looking into Power Platform and Power BI, office 365 for the end 2 end AI Tech to Business experience to enable the Business users to do more with AI support and collaborate together for successful outcomes. Are there other offerings that we could / should be focused on for AI advantage? (link)

A: This AMA is targeted for questions around Azure Cognitive Search so I don't think we'll be able to answer this question for you. You could try posting your question in the Power Platform Community.

Q: When using Azure OpenAI on your data, do the search terms retrieved from Azure Cognitive Search count toward the number of tokens and thus to the charge for Azure OpenAI? If so, is there any way for customers to predict or estimate the costs of using Azure OpenAI on your data? (link)

A: Yes, whatever is retrieved from Azure Cognitive Search and injected into the Azure OpenAI Prompt will count towards the token count (and thus the cost to execute that Azure OpenAI request. Pablo created a really good blog post that goes into more details on this. This is ultimately why the relevance of Azure Cognitive Search is so important and why we spend a lot of our time in making sure we can get the best results in the top 3-5 results to help minimize the number of tokens that are required. As for the cost, you will typically be "chunking" your data which should give you a good idea on the cost of any Azure OpenAI request as you can look at the typical number of tokens per chunk x the number of results you add to the prompt.

If you are only using vector search to retrieve documents, and not passing those documents to an Azure OpenAI prompt, then the usage of vector search by itself will not result in any calls to Azure OpenAI. Meaning, once you've generated the vector embeddings for your content (which will consume token counts in the embedding model/endpoint), there are no further costs beyond the search service itself. It's only when you pass the retrieved documents where additional costs can come into play.

Q: What is the road map for vector search and what improvements and features are planned for future. It is in public preview right now, when is the plan to make it prod ready. Will you release RAG as service? where we can have configurations like embedding model endpoint and input store connection. (link)

A: For the RAG as a service, I would highly recommend you take a look at Azure OpenAI on your Data as well as the related blog post, which I think is likely very close to what you might be looking for which is a really easy way to get up and running with RAG and enterprise ChatGPT. As for futures for vector search, our main goals are to simplify a lot of the complexities that come with vectorization of content which include looking at how to simplify ingestion and queries when vectorization is required as well as topics such as effective chunking of data.

Q: To use Snowflake as a data source, are you aware of an officially-supported module or do you have a recommended shim approach to use one of the published sources? (https://learn.microsoft.com/EN-US/AZURE/search/search-data-sources-gallery) (link)

A: One good option is to use Azure Data Factory, which has a Snowflake connector. to ingest data. Using this, you can also take the data copied and send it to Azure Cognitive Search. It is however important to note that you can also do this programmatically if you prefer using our PUSH API which allows you to format your data in JSON format and send it directly to Azure Cognitive Search.

Q: Is Vector Search going to have its own ranking? like Semantic ranking? (link)

A: Could you explain what you mean by vector search ranking? Semantic ranking is a re-ranking model so it's slightly different.

There is new ranking for vector search: vector search queries can be ranked using cosine similarity, Euclidean distance, or dot products. Azure Cognitive Search also supports hybrid search which combines keyword based search and vector search results using reciprocal rank fusion.

This document has a great overview of how ranking works with vector search: Query vector data in a search index - Azure Cognitive Search | Microsoft Learn

Q: We have implemented azure cognitive search for finding products based on their attributes and this works very well. Now a customer has the requirement to be able to search products based on the ETIM classification attributes. This classification has a predefined number of over 14.000 unique attributes across all product categories. As each unique attribute will consume at least one simple field we run into the index limit of a 1000. Maybe we have chosen the wrong architecture, but what is the right one to use when we have such a requirement? (link)

A: This appears to be a question about the 1000 field limit in Azure Cognitive Search. There isn't an easy workaround to the field limit, but you do have some options to get the experience you are looking for.

These are 3 options I would suggest considering:

Field reuse within an index. If each product catalog entry would use less than 1000 fields, you could reuse the same fields for multiple product types with your application managing the mapping of the meaning of each attribute to the underlying field in the index. This option does have drawbacks since each field would have the same settings across multiple product types.

Extend catalog to multiple indexes. You could effectively store your catalog in multiple indexes where the full set of fields will extend across the indexes. This would work best if an individual catalog entry doesn't have more than 1000 fields again and would run into some cross-index query issues. But you should be able to get faceting to work in this option.

Attribute metadata fields. Define a few general fields in your index to store attribute metadata. These fields could include "attribute_name," "attribute_value," and "product_id" or any other relevant identifiers. For each product, store its ETIM classification attributes as separate records in the attribute metadata fields. This means you'll have multiple attribute records for each product, one for each ETIM attribute associated with that product. To implement this solution, you'll need to update your data indexing and search query pipelines to work with the attribute metadata fields and apply filters accordingly. It may require some adjustments to your existing code, but it should provide a way to handle the complex attribute requirements.

Q: How does ACS compare to Redis for vector Search. In terms of cost and performance. We have experienced better performance with Redis. Any plans to improve performance and can you share your views. How does search performance change when the DB size increases? (link)

A: Vector search performance is a very interesting topic. The Approximate Nearest Neighbors methods, being approximate, use different approaches to influence the speed/recall tradeoff. In the limit, you can achieve perfect recall by comparing the query vector to all vectors in the database but that's obviously too expensive. To improve speed, ANN algorithms leverage compression techniques lower precision, or data structures to partition the indexed data to reduce the number of vector comparisons that need to be made. All of it to say that statements about performance should only be made at a specific recall target. Another dimension to this problem is price since you can choose to spend more on hardware to improve search latency and throughput. In an ideal benchmark, you'd pick configurations comparable on price and recall before you look at search performance since they are all related.

Both Cognitive Search and Redis use HNSW as the Approximate Nearest Neighbors algorithm which is one of the leading methods for applications optimizing for low latency and high recall and data that's indexed incrementally and can change over time. The main differences in price/performance could stem from differences in implementation and other functionalities the service offers in addition to Vector search which are included in the price i.e., scaling, security, compliance, integration with other services, etc.

Hope this helps you reason about Vector search performance in Azure Cognitive Search and how it compares. We're planning to publish results of our benchmarks which compares Vector search performance between different Cognitive Search SKUs and service topologies – how adding replicas and partitions changes Vector search performance profile. The basic intuition is that adding replicas improves throughput and adding partitions reduces latency.

Q: Not related to RAG but is there any plans to add Function Calling to Azure OpenAI? (link)

A: Good news: Function calling is now available in Azure OpenAI Service - Microsoft Community Hub

Q: For some new to AI and Cognitive Search, what skills should you learn to get a job ? are those skills on MS Learn :-)? Thank you in advance for guidance!!!! :) (link)

A: Microsoft does have some relevant modules on MS Learn to help you boost your skills. Here are a few courses that you could consider:

- Microsoft Azure AI Fundamentals: Get started with artificial intelligence - Training | Microsoft Learn

- Introduction to Azure Cognitive Search - Training | Microsoft Learn

- Introduction to Azure OpenAI Service - Training | Microsoft Learn

Q: I'm interested in information or resources on what the Retrieval Augmented Generation pattern with Cognitive Search and vector search looks like end-to-end. Specifically comparing it to the existing samples available in GitHub would be extra useful. (link)

A: I think this blog post that Pablo Castro wrote is the best resource to get started here and to learn more about RAG and the Retrieve Augment Generate pattern. From there, the demo code which can be found here. is a great more technical resource.

Q1: That is the pattern I have been following in my experimentation using the sample code from GitHub as well. How will vector search change or improve upon this?

A1: RAG pattern consists broadly of two steps:

[the summary of RAG pattern below, number points 1 and 2 are sourced from https://vitalflux.com/retrieval-augmented-generation-rag-llm-examples/]

1. Retrieval Phase: Given an input query (like a question), the RAG system first retrieves relevant documents or passages from a large corpus using a retriever. This is often done using efficient dense vector space methods, like the Dense Retriever (DPR), which embeds both the query and documents into a continuous vector space and retrieves documents based on distance metrics.

2. Generation Phase: Once the top-k relevant documents or passages are retrieved, they are fed into a sequence-to-sequence generator along with the original query. The generator is then responsible for producing the desired output (like an answer to the question) using both the query and the retrieved passages as context.

During the retrieval phase, the candidate documents that are returned will directly affect the generation phase, as the quality of that phase will only be as good as the input documents and the completion model.

As a result, it is beneficial to improve the quality of the retrieval phase. This is where vector search can improve on this RAG pattern. In many common scenarios, vector search can return more semantically relevant information than traditional keyword search, because it is searching based on the meaning of the search query and candidate documents and doesn't require keyword matches and term frequency, document length, term saturation, etc. that TF IDF and BM25 keyword search techniques would use. This brings all the powerful capabilities of vector search, such as multi-lingual, multi-modal, etc. to surface potentially more relevant documents.

(Please don't confuse the use of the word 'semantic' above with Azure Cognitive Search feature, "semantic search". Here I'm only using semantic as an adjective.)

Q: When will the documentation on the Semantic Index and how to use it be released? (link)

A: You can find documentation about semantic search in Azure Cognitive Search here: https://learn.microsoft.com/en-us/azure/search/semantic-search-overview When you say "Semantic Index", is this what you are referring to?

Q1: No, Semantic Index is, according to the docs, what Msft Co-Pilots will be using. My understanding is it will be available for customer use (E3 and E5) as well but have not seen any documentation. My working assumption is it will be exposed via a graph connector perhaps.

A1: In the context of this AMA, we are only from the Azure Cognitive Search Product Group and don't have any insight into what the MSFT Co-Pilots will be using nor their release plans and documentation.

Q: It seems vector search will usually outperform a text-based semantic search in terms of accurate results. Do you have any hard data showing a side by side comparison of these 2 search methods? (link)

A: Vector search and Semantic search are somewhat orthogonal but aim to serve the same function – improve the quality of the search results. In this context, Vector search is used to improve recall – the number of relevant results returned by a search query. Semantic search improves ranking by bringing up the most relevant documents to the top of the results list. Vector search and Semantic search can be used independently but work best when used together. Note, Semantic search today works only for search requests that include a text query. The effectiveness of each method will depend on your scenario. For example, Vector search is

only as good as the model you use to vectorize the data and you should refer to the model benchmarks to understand its strengths and weaknesses. Semantic search leverages Bing models trained on the web corpus, so it's very effective for data that's not domain-specific, but it's best to test yourself on your data (Semantic search is free for the first 1000 requests).

Q: Will "auto-vectorization" with an embedding model of choice (AOAI's ada or HF's sentence transformers), including intelligent splitting/chunking, be an option for any of Azure's persistence providers or ACS? (link)

A: Please stay tuned for such functionality in the upcoming months. It is currently in our roadmap.

Q: I'd love to gain some insight into the pricing model of vector search. Based on the limited information available on the semantic feature in Cognitive Search, I'm guessing there will be some base cost of ~$500 per month. Why the base cost component? Also, what will be the price be approximately per token/document ingested? Thanks! (link)

A: That is correct, unfortunately (or fortunately), the semantic model leverages GPU's that has a cost associated with it. We do understand that it makes more sense for some customers to have a more gradual ("pay per usage") type model as opposed to having the $500 cost for a large number of transactions. Please keep an eye out for that. It is important to note that for the semantic search capability, there is no cost for ingesting data as it is purely a reranking model. More details can be found here: https://learn.microsoft.com/en-us/azure/search/semantic-search-overview

A2: Adding to Liam's answer, using semantic search along with vector search is not a requirement to use vector search features itself. Customers may often find that semantic search will improve the relevance and ranking of the search results, so you may wish to perform testing on whether to use semantic search if you are more sensitive to the base cost.

If you are only using vector search, there are no additional costs to use this feature within your pre-existing search service. Please note the service limits and restrictions outlined in these pages below:

Vector limits: https://learn.microsoft.com/en-us/azure/search/search-limits-quotas-capacity#vector-index-size-limits

Vector availability and pricing: https://learn.microsoft.com/en-us/azure/search/search-limits-quotas-capacity#vector-index-size-limits

Q: In your Blog (https://techcommunity.microsoft.com/t5/azure-ai-services-blog/announcing-vector-search-in-azure-cognitive-search-public/ba-p/3872868) you are referring to Vector Search (using embeddings) for Audio. Which Services / APIs are provided for that? (link)

A: That somewhat depends on what the use case is. For example, a very common one is to be able to convert audio to text (e.g., transcription) which is then searchable. For this there is Azure

Speech Services (https://learn.microsoft.com/en-us/azure/ai-services/speech-service/overview) as well as the introduction of OpenAI Whisper (https://techcommunity.microsoft.com/t5/azure-ai-services-blog/openai-whisper-is-coming-soon-to-azure-openai-service-and-azure/ba-p/3876671). Once it is in text format you can then use typical text models such as Azure OpenAI Ada 002. There are also numerous models that can be used to create embeddings based on other audio types.

Q: Looking forward to this session. I have a few questions that it would be great to have covered:

I'm using the RAG pattern currently with ACS semantic search to find relevant content to include in chat prompts. What is the difference that vector search will bring and when would you choose one over the other?

With vector search, will vectorization of content need to be done externally before passing into ACS?

Are there/will there be updated samples to show the power of vector search in ACS?

CosmosDB for MongoDB Core offers vector search - any guidance on when you might choose ACS vs CosmosDB for that capability?

Thanks in advance! (link)

A: Vector search is a new feature in Azure Cognitive Search that is currently in public preview. It is designed to provide more advanced search capabilities by using vectorization techniques to represent documents and queries as vectors in a high-dimensional space. This allows for more efficient and accurate matching of similar documents and queries based on their semantic meaning and context. Vector search can be used in conjunction with the RAG pattern to provide more advanced question-answering capabilities and improve the accuracy of search results. With vector search, you would need to perform vectorization of content externally before passing it into Azure Cognitive Search. This can be done using various techniques, such as word embeddings or deep learning models, depending on the specific requirements of your scenario. There are updated samples available that demonstrate the power of vector search in Azure Cognitive Search. You can refer to the Azure Cognitive Search documentation and the SDK documentation for code samples and guidance on vector search implementation and management. Regarding CosmosDB for MongoDB Core offering vector search, it's important to evaluate the specific requirements and constraints of your scenario to determine which solution is best suited for your needs. Azure Cognitive Search provides a fully managed, scalable, and flexible search service that can handle various types of data and workloads. It also integrates seamlessly with other Azure services, such as Azure OpenAI Service, to provide more advanced natural language processing capabilities. CosmosDB for MongoDB Core provides a fully managed NoSQL database service that can handle various types of data and workloads. It also provides built-in support for MongoDB APIs and features, such as sharding and replication.

A1: Thanks for the great response! I would like to add that Azure Cognitive Search not only offers Vector Search, but also Hybrid Search which leverages scores from traditional text search as well as vectors, which we (and much of the industry research) has found to offer more effective relevance than just Vector Search. In addition, when you then add our Semantic Search (which is a reranking layer), we find this typically offers the most effective relevance, which is incredibly important, especially when build enterprise ChatGPT apps. We are working on a blog post around the effectiveness of this, so please keep your eye out over the next few weeks here.

Q: What information is stored within the Open AI Service. If a user asks a question or provides information that includes PII data or special category data that fall under GDPR regulations, how is that data handled? Is the data used to train the system? Is there any way this type of data could be redacted when stored? (link)

A: Please take a look at this link for more details on data, privacy, and security with the Azure OpenAI Service.

A1: Unlike the ChatGPT website, data that is sent to the Azure OpenAI API endpoints is not (by default) used for Reinforcement Learning from Human Feedback (RLHF) and as such does not get added back to the GPT foundational models.

To use Azure OpenAI, you first need to create that resource within the Cognitive Services blade along with selecting the region. Data processed in that region meets the same regulatory requirements as other services within that region. Beyond this scaffolding, it's up to the resource owners to build services that meet applicable regulatory requirements. Azure also provides the same type of logs for Azure OpenAI as it does other services which leverage Azure Monitor.

Reference URLs:

How your data is used to improve model performance | OpenAI Help Center

Azure Cognitive Services security - Azure Cognitive Services | Microsoft Learn

Monitoring Azure OpenAI Service - Azure Cognitive Services | Microsoft Learn

Q: Currently the Azure Open AI Service is not available in a UK region. When will it become available in a UK region? (link)

A: It's available now. You can deploy gpt-35-turbo in "UK South" Azure region as of yesterday: https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/models#gpt-3-models-1.

Q: Do we have any best practices or case studies for beginners to learn Azure OpenAI & Cognitive Search? (link)

A: For Azure Cognitive Search, a good hands-on holistic overview of the whole product is here Knowledge Mining Accelerator, and specifically for Vector Search, Quickstart vector search - Azure Cognitive Search | Microsoft Learn.

For Azure OpenAI, a good start is Quickstart - Deploy a model and generate text using Azure OpenAI Service - Azure OpenAI | Microsoft Learn

Q: How do you manage the cost of using Azure Search as vector store for RAG use cases? I had an experience trying out an azure search example that involves some 90k arvix and covid journals and it costs £500 just to load them. Would there be a way to estimate cost better and how should one approach it? (link)

A: Cost ultimately comes down to your SKU and the number of Search Units (SUs) which many factors play a role in the decision making such as document count, availability and reliability, index schema design, count of vector fields, vector field dimensions, etc. The most accurate way to estimate cost is to perform a proof-of-concept index with the desired schema you want and load a sample of documents that will reflect your production workload. Then, you can extrapolate your sample document count and index size with your production document count and get your production index size. You can visit our documentation on service limits by SKU and decide which SKU and Search Unit Count (partitions/replicas) fit your needs best. https://learn.microsoft.com/azure/search/search-limits-quotas-capacity Once you have an estimated SKU and number of search units, you can visit our pricing calculator here, select your region, and see a cost estimate of your Cognitive Search service: https://azure.microsoft.com/pricing/details/search/

Q: We are in technical roles working to help deliver some prototype AI features into our application suite.

Some topics on our team's mind in preparation for the live session:

Multi-tenancy 1: Are there any formal recommendations on having multi-tenant Cognitive Search-LLM via Azure AI Studio? (beyond having a full instance per tenant)

Multi-tenancy 2: We are proofing the idea of having multiple indexes in a single cognitive search resource - one for each of our customers. We would then have a single LLM that would process the prompt along with the results of the particular index search based on the customer. Are there any limits to the number of indexes within one Search resource? Are there any risks or challenges we should be aware of in using this approach?

In all of the samples, the pattern leverages a Blob Container with documents that are indexed with the index being automatically set up by the Open AI Studio. We are wondering how we would do that from a straight code/automation perspective. What are the commands/sdk that we use to create a new index for a Blob Container that pulls out the correct 5 pieces of metadata?

Since Azure Cognitive Search can handle databases and json data - Does Search + Azure OpenAI also support pure data from SQL Server or json documents? Or are documents (Word, PDF, etc.) the only items supported in that scenario?

What is the difference between the regular search and the higher priced semantic search with regards to the RAG pattern? (link)

A: I can help with #5.

What is the difference between the regular search and the higher priced semantic search with regards to the RAG pattern?

RAG pattern consists broadly of two steps:

[the summary of RAG pattern below, number points 1 and 2 are sourced from https://vitalflux.com/retrieval-augmented-generation-rag-llm-examples/]

1. Retrieval Phase: Given an input query (like a question), the RAG system first retrieves relevant documents or passages from a large corpus using a retriever. This is often done using efficient dense vector space methods, like the Dense Retriever (DPR), which embeds both the query and documents into a continuous vector space and retrieves documents based on distance metrics.

2. Generation Phase: Once the top-k relevant documents or passages are retrieved, they are fed into a sequence-to-sequence generator along with the original query. The generator is then responsible for producing the desired output (like an answer to the question) using both the query and the retrieved passages as context.

During the retrieval phase, the candidate documents that are returned will directly affect the generation phase, as the quality of that phase will only be as good as the input documents and the completion model.

As a result, it is beneficial to improve the quality of the retrieval phase. This is where vector search and/or semantic search can improve on this RAG pattern. Both features (either used together or only using one or the other) can return more semantically relevant information than traditional keyword search, because they are searching based on the meaning of the search

query and candidate documents and doesn't require keyword matches and term frequency, document length, term saturation, etc. that TF IDF and BM25 keyword search techniques would use.

A2 (non-employee): For multi-tenancy, Azure Cognitive Search has a few common patterns when modeling a multitenant scenario. One index per tenant: Each tenant has its own index within a search service that is shared with other tenants. One service per tenant: Each tenant has its own dedicated Azure Cognitive Search service, offering the highest level of data and workload separation.

Regarding multiple indexes in one resource, Azure Cognitive Search can import, analyze, and index data from multiple data sources into a single consolidated search index. You can use multiple indexers in Azure Cognitive Search to create a single search index from files in Blob storage, with additional file metadata in Table storage. You can also configure an indexer that imports content from Azure Blob Storage and makes it searchable in Azure Cognitive Search.

To create an index for a Blob Container that pulls out the correct 5 pieces of metadata, you can use the deploy-index.json file which defines the structure of the search index. It includes the typical information from blob storage (the content as well as file name, full path, file size, etc.).

Azure Cognitive Search supports pure data from SQL Server or JSON documents. It also supports documents (Word, PDF, etc.).

The difference between regular search and semantic search with regards to the RAG pattern is that semantic search uses natural language processing (NLP) to understand the meaning behind words and phrases. It can identify synonyms and related concepts to expand queries and improve relevance. Regular search uses keyword matching to find relevant documents.

Q: Is there a way with Cogsearch to pick up the purview data classification tags when documents are index so when a prompt reply contains info from a document tagged restricted confidential the LLM/Response can embed in the response - data may be Restricted confidential with 100% reliability? (link)

A: You can leverage the Cognitive Search Indexer feature, to index documents from different data sources. https://learn.microsoft.com/azure/search/search-indexer-overview

You can use the filter predicate pattern for security filter trimming in Cognitive Search, ensuring that results retrieved from your prompt are only accessible to users with access.
See https://learn.microsoft.com/azure/search/search-security-trimming-for-azure-search.

I encourage you to test and evaluate these solutions thoroughly if you have company confidential or restricted data before launching to production.

That's a wrap!

Thank you for joining this fun hour! We hope you'll continue to ask questions and share your feedback.

See you next time!