

Implementación de la minería de conocimiento con Búsqueda de Azure AI

¿Tiene información bloqueada en orígenes de datos estructurados y no estructurados? Con Búsqueda de Azure AI, puede extraer información clave de estos datos y permitir que las aplicaciones la busquen y analicen.

Requisitos previos

Familiaridad con Microsoft Azure

Experiencia de desarrollo de aplicaciones con C# o Python

Introducción

Todas las organizaciones se basan en información para tomar decisiones, responder preguntas y operar de manera eficaz. El problema de la mayoría de las organizaciones no es una falta de información, sino el reto de buscar y extraer la información del conjunto masivo de documentos, bases de datos y otros orígenes en los que se almacena.

Por ejemplo, supongamos que Margie's Travel es una agencia de viajes especializada en organizar viajes a ciudades de todo el mundo. Con el tiempo, la empresa ha acumulado una enorme cantidad de información en documentos, como folletos, y reseñas de hoteles enviadas por los clientes. Estos datos son una valiosa fuente de información para los agentes de viajes y los clientes a medida que planifican los viajes, pero el volumen ingente de datos puede dificultar la búsqueda de información pertinente para responder preguntas específicas de los clientes.

Para abordar este reto, Margie's Travel puede implementar una solución en la que los documentos están indexados y es más fácil buscar en ellos. Esta solución permite a agentes y clientes consultar el índice para encontrar documentos pertinentes y extraer información de ellos.

Azure AI Search

Búsqueda de Azure AI proporciona una solución basada en la nube para indexar y consultar una amplia gama de orígenes de datos, así como para crear soluciones de búsqueda completas y a gran escala. Con Búsqueda de Azure AI, puede hacer lo siguiente:

Indexar documentos y datos de una variedad de orígenes.

Usar aptitudes cognitivas para enriquecer datos de índice.

Almacenar conclusiones extraídas en un almacén de conocimiento para su análisis e integración.

Al final de este módulo, sabrá cómo:

Creación de una solución de Búsqueda de Azure AI

Desarrollar una aplicación de búsqueda

Administración de capacidad

Para crear una solución de Búsqueda de Azure AI, debe crear un recurso de Búsqueda de Azure AI en la suscripción de Azure. Dependiendo de la solución específica que quiera crear, puede que también necesite recursos de Azure para el almacenamiento de datos y otros servicios de aplicación.

Niveles de servicio y administración de la capacidad

Al crear un recurso de Búsqueda de Azure AI, debe especificar un plan de tarifa. El plan de tarifa que seleccione determinará las limitaciones de capacidad del servicio de búsqueda y las opciones de configuración disponibles, así como el costo del servicio. Los planes de tarifa disponibles son:

Gratis (F): use este plan para explorar el servicio o probar los tutoriales de la documentación del producto.

Básico (B): Use este nivel para soluciones de búsqueda a pequeña escala que incluyan un máximo de 15 índices y 5 GB de datos de índice.

Estándar (S): use este plan para soluciones a escala empresarial. Hay varias variantes de este plan, como S, S2 y S3, que ofrecen una mayor capacidad en términos de índices y almacenamiento, y S3HD, que está optimizado para un rendimiento de lectura rápido en un número menor de índices.

Almacenamiento optimizado (L): use un plan de almacenamiento optimizado (L1 o L2) cuando necesite crear índices de gran tamaño, a expensas de una mayor latencia de consulta.

Réplicas y particiones

Según el plan de tarifa que seleccione, puede optimizar su solución para aumentar la escalabilidad y disponibilidad mediante la creación de réplicas y particiones.

Las réplicas son instancias del servicio de búsqueda; puede considerarlas como nodos de un clúster. Aumentar el número de réplicas puede ayudar a garantizar que haya suficiente capacidad para atender varias solicitudes de consulta simultáneas mientras se administran las operaciones de indexación en curso.

Las particiones se usan para distribuir un índice en varias ubicaciones de almacenamiento, lo que permite dividir las operaciones de E/S, como realizar consultas o volver a generar un índice.

La combinación de réplicas y particiones que configure determinará las unidades de búsqueda que usa la solución. En pocas palabras, el número de unidades de búsqueda es el número de réplicas multiplicado por el número de particiones ($R \times P = SU$). Por ejemplo, un recurso con cuatro réplicas y tres particiones usa 12 unidades de búsqueda.

Comprender los componentes de búsqueda

Una solución de Búsqueda de AI consta de varios componentes, y cada uno desempeña un papel importante en el proceso de extracción, enriquecimiento, indexación y búsqueda de datos.

Origen de datos

La mayoría de las soluciones de búsqueda comienzan con un origen de datos que contiene los datos que se quieren buscar. Búsqueda de Azure AI admite varios tipos de origen de datos, como los siguientes:

Archivos no estructurados en contenedores de Azure Blob Storage.

Tablas de Azure SQL Database.

Documentos de Cosmos DB.

Búsqueda de Azure AI puede extraer datos de estos orígenes de datos para la indexación.

Como alternativa, las aplicaciones pueden insertar datos JSON directamente en un índice, sin extraerlos de un almacén de datos existente.

Conjunto de aptitudes

En una solución de búsqueda básica, puede indexar los datos extraídos del origen de datos. La información que se puede extraer depende del origen de datos. Por ejemplo, al indexar datos de una base de datos, es posible que se extraigan los campos de las tablas de la base de datos; o bien, al indexar un conjunto de documentos, es posible que se extraigan los metadatos del archivo, como el nombre de archivo, la fecha de modificación, el tamaño y el autor, junto con el contenido de texto del documento.

Aunque una solución de búsqueda básica que indexa los valores de datos extraídos directamente desde el origen de datos resulta útil, las expectativas de los usuarios de aplicaciones modernas han impulsado una necesidad de obtener conclusiones más completas a partir de los datos. En Búsqueda de Azure AI, puede aplicar aptitudes de inteligencia artificial (IA) como parte del proceso de indexación para enriquecer los datos de origen con nueva información, que se puede asignar a campos de índice. Las aptitudes usadas por un indexador se encapsulan en un conjunto de aptitudes, el cual define una canalización de enriquecimiento en la que cada paso mejora los datos de origen con conclusiones obtenidas por una aptitud de IA específica. Entre los ejemplos del tipo de información que puede extraer una aptitud de IA se incluyen:

El lenguaje en el que está escrito un documento.

Frases clave que pueden ayudar a determinar los temas principales que se tratan en un documento.

Una puntuación de opinión que indica lo positivo o negativo que es un documento.

Ubicaciones, personas, organizaciones o puntos de referencia específicos mencionados en el contenido.

Descripciones de imágenes generadas por IA o texto de imágenes extraído mediante el reconocimiento óptico de caracteres.

Aptitudes personalizadas que desarrolla para satisfacer requisitos específicos.

Indizador

El indexador es el motor que impulsa el proceso de indexación general. Toma las salidas extraídas mediante las aptitudes del conjunto de aptitudes, junto con los valores de datos y metadatos extraídos del origen de datos original, y las asigna a campos del índice.

Un indexador se ejecuta automáticamente cuando se crea y se puede programar para ejecutarse a intervalos regulares o a petición para agregar más documentos al índice. En algunos casos, como cuando se agregan campos nuevos a un índice o aptitudes nuevas a un conjunto de aptitudes, es posible que tenga que restablecer el índice antes de volver a ejecutar el indexador.

Índice

El índice es el resultado que se puede buscar del proceso de indexación. Consta de una colección de documentos JSON, con campos que contienen los valores extraídos durante la indexación. Las aplicaciones cliente pueden consultar el índice para recuperar, filtrar y ordenar información.

Cada campo de índice se puede configurar con los atributos siguientes:

key: campos que definen una clave única para los registros del índice.

searchable: campos que se pueden consultar mediante la búsqueda de texto completo.

filterable: campos que se pueden incluir en expresiones de filtro para devolver solo los documentos que coincidan con las restricciones especificadas.

sortable: campos que se pueden usar para ordenar los resultados.

facetable: campos que se pueden usar para determinar los valores de las facetas (elementos de la interfaz de usuario que se usan para filtrar los resultados en función de una lista de valores de campo conocidos).

retrievable: campos que se pueden incluir en los resultados de la búsqueda (de forma predeterminada, todos los campos se pueden recuperar a menos que este atributo se quite explícitamente).

Descripción del proceso de indexación

El proceso de indexación funciona mediante la creación de un documento para cada entidad indizada. Durante la indexación, una canalización de enriquecimiento crea de forma iterativa los documentos que combinan metadatos del origen de datos con campos enriquecidos extraídos por las aptitudes cognitivas. Cada documento indizado se puede considerar como una estructura JSON, que inicialmente consta de un documento con los campos del índice que ha asignado a los campos extraídos directamente de los datos de origen; por ejemplo:

document

metadata_storage_name

metadata_author

content

Cuando los documentos del origen de datos contienen imágenes, puede configurar el indexador para que extraiga los datos de las imágenes y coloque cada imagen en una colección `normalized_images`; por ejemplo, de la siguiente manera:

document

metadata_storage_name

metadata_author

content

normalized_images

image0

image1

La normalización de los datos de las imágenes de esta manera le permite usar la colección de imágenes como una entrada para las aptitudes que extraen información a partir de datos de imagen.

Cada aptitud agrega campos al documento, por lo que, por ejemplo, una aptitud que detecta el idioma en el que está escrito un documento podría almacenar su salida en un campo language de la siguiente manera:

document

metadata_storage_name

metadata_author

content

normalized_images

image0

image1

language

El documento está estructurado jerárquicamente y las aptitudes se aplican a un contexto específico dentro de la jerarquía, lo que le permite ejecutar la aptitud para cada elemento en un nivel determinado del documento. Por ejemplo, podría ejecutar una aptitud de reconocimiento óptico de caracteres (OCR) para cada imagen de la colección de imágenes normalizadas a fin de extraer el texto que contengan:

document

metadata_storage_name

metadata_author

content

normalized_images

image0

Texto

image1

Texto

language

Los campos de salida de cada aptitud se pueden usar más adelante en la canalización como entradas para otras aptitudes que, a su vez, almacenan sus salidas en la estructura del documento. Por ejemplo, se podría usar una aptitud de combinación para combinar el contenido de texto original con el texto extraído de cada imagen para crear un campo merged_content que contenga todo el texto del documento, incluido el de la imagen.

document

metadata_storage_name

metadata_author

content

normalized_images

image0

Texto

image1

Texto

language

merged_content

El indexador asigna los campos de la estructura final del documento al final de la canalización a los campos de índice de una de estas dos maneras:

Los campos extraídos directamente de los datos de origen se asignan a campos de índice. Estas asignaciones pueden ser implícitas (los campos se asignan automáticamente en campos con el mismo nombre en el índice) o explícitas (se define una asignación para hacer coincidir un campo de origen con un campo de índice, a menudo para cambiar el nombre del campo por algo más útil, o bien para aplicar una función al valor de datos a medida que se asigna).

Los campos de salida de las aptitudes del conjunto de aptitudes se asignan de forma explícita desde su ubicación jerárquica en la salida al campo de destino del índice.

Búsqueda de un índice

Después de crear y rellenar un índice, puede consultarlo para buscar información en el contenido de los documentos indexados. Aunque podría recuperar entradas de índice en función de la coincidencia simple con el valor del campo, la mayoría de las soluciones de búsqueda usan la semántica de búsqueda de texto completo para consultar un índice.

Búsqueda de texto completo

La búsqueda de texto completo describe soluciones de búsqueda que analizan el contenido de documentos basados en texto para buscar términos de consulta. Las consultas de búsqueda de texto completo en Búsqueda de Azure AI se basan en la sintaxis de consulta de Lucene, que proporciona un amplio conjunto de operaciones de consulta para buscar, filtrar y ordenar datos en índices. Búsqueda de Azure AI admite dos variantes de la sintaxis de Lucene:

Simple: una sintaxis intuitiva que facilita la realización de búsquedas básicas de coincidencias con los términos de consulta literales enviados por un usuario.

Full: una sintaxis extendida que admite el filtrado complejo, expresiones regulares y otras consultas más sofisticadas.

Las aplicaciones cliente envían consultas a Búsqueda de Azure AI especificando una expresión de búsqueda junto con otros parámetros que determinan cómo se evalúa la expresión y los resultados devueltos. Algunos de los parámetros comunes que se envían con una consulta son los siguientes:

search: una expresión de búsqueda que incluye los términos que se van a buscar.

queryType: la sintaxis de Lucene que se va a evaluar (simple o full).

searchFields: los campos del índice en los que se va a buscar.

select: los campos que se van a incluir en los resultados.

searchMode: los criterios para incluir resultados basados en varios términos de búsqueda. Por ejemplo, supongamos que busca un hotel cómodo. Un valor **searchMode Any** devuelve los documentos que contienen "cómodo", "hotel" o ambos términos. Mientras que un valor **searchMode All** restringe los resultados a los documentos que contienen tanto "cómodo" como "hotel".

El procesamiento de consultas se compone de cuatro fases:

Análisis de consultas: la expresión de búsqueda se evalúa y se reconstruye como un árbol de subconsultas adecuadas. Las subconsultas pueden incluir consultas de términos (para buscar palabras individuales específicas en la expresión de búsqueda, como hotel), consultas de frases (para buscar frases con varios términos especificados entre comillas en la expresión de búsqueda, como "aparcamiento gratuito") y consultas de prefijo (para buscar términos con un prefijo especificado; por ejemplo, **aero***, que devolvería coincidencias como **aerobic**, **aerolínea** y **aeropuerto**).

Análisis léxico: los términos de consulta se analizan y refinan según las reglas lingüísticas. Por ejemplo, el texto se convierte a minúsculas y se quitan las palabras irrelevantes no esenciales (como "el", "un", "es", etc.). A continuación, las palabras se convierten en su forma raíz (por ejemplo, "confortable" podría simplificarse como "confort") y las palabras compuestas se dividen en sus términos constituyentes.

Recuperación de documentos: los términos de consulta se comparan con los términos indexados y se identifica el conjunto de documentos coincidentes.

Puntuación: se asigna una puntuación de relevancia a cada resultado en función de un cálculo de la frecuencia del término y la frecuencia inversa del documento (TF/IDF).

Filtrado y ordenación de los datos

Es habitual en una solución de búsqueda que los usuarios quieran restringir los resultados de las consultas mediante el filtrado y la ordenación en función de los valores de los campos. Búsqueda de Azure AI es compatible con estas dos funcionalidades a través de la API de consulta de búsqueda.

Filtrado de los resultados

Puede aplicar filtros a las consultas de dos maneras:

Mediante la inclusión de criterios de filtro en una expresión de búsqueda simple.

Proporcionando una expresión de filtro de OData como un parámetro \$filter con una expresión de búsqueda de sintaxis completa (full).

Puede aplicar un filtro a cualquier campo filtrable (filterable) del índice.

Por ejemplo, supongamos que quiere buscar documentos que contienen el texto London (Londres) y que tienen un campo author (autor) con el valor Reviewer (Revisor).

Puede lograr este resultado enviando la siguiente expresión de búsqueda simple:

```
search=London+author='Reviewer'
```

```
queryType=Simple
```

Como alternativa, puede usar un filtro de OData en un parámetro \$filter con una expresión de búsqueda de Lucene completa (full) como la siguiente:

```
search=London
```

```
$filter=author eq 'Reviewer'
```

```
queryType=Full
```

Filtrado con facetas

Las facetas son una manera útil de presentar a los usuarios criterios de filtrado basados en valores de campo en un conjunto de resultados. Funcionan mejor cuando un campo tiene un pequeño número de valores discretos que se pueden mostrar como vínculos u opciones en la interfaz de usuario.

Para usar facetas, debe especificar los campos clasificables (facetable) para los que quiere recuperar los valores posibles en una consulta inicial. Por ejemplo, puede usar los parámetros siguientes para devolver todos los valores posibles para el campo author:

```
search=*
```

```
facet=author
```

Los resultados de esta consulta incluyen una colección de valores de faceta discretos que puede mostrar en la interfaz de usuario para que el usuario los seleccione. Después, en una consulta posterior, puede usar el valor de faceta seleccionado para filtrar los resultados:

```
search=*
```

```
$filter=author eq 'selected-facet-value-here'
```

Ordenar resultados

De forma predeterminada, los resultados se ordenan en función de la puntuación de relevancia asignada por el proceso de consulta, con las coincidencias de puntuación más alta enumeradas primero. Pero puede invalidar este criterio de ordenación incluyendo un parámetro orderby de

OData que especifique uno o más campos ordenables (sortable) y un criterio de ordenación (ascendente o descendente).

Por ejemplo, para ordenar los resultados de forma que los documentos modificados más recientemente aparezcan en primer lugar, puede usar los siguientes valores de parámetro:

`search=*`

`$orderby=last_modified desc`

Mejora del índice

Con un índice básico y un cliente que puede enviar consultas y mostrar resultados, puede lograr una solución de búsqueda eficaz. Sin embargo, la Búsqueda de Azure AI es compatible con varias formas de mejorar un índice para ofrecer una mejor experiencia al usuario. En este tema se describen algunas de las formas en que se puede ampliar la solución de búsqueda.

Búsqueda mientras se escribe

Al agregar un proveedor de sugerencias a un índice, puede habilitar dos formas de experiencia de búsqueda mientras se escribe para ayudar a los usuarios a encontrar resultados pertinentes más fácilmente:

Sugerencias: recupere y muestre una lista de resultados sugeridos cuando el usuario escriba en el cuadro de búsqueda, sin necesidad de enviar la consulta de búsqueda.

Autocompletar: complete términos de búsqueda escritos parcialmente en función de los valores de los campos de índice.

Para implementar una o ambas capacidades, cree o actualice un índice definiendo un proveedor de sugerencias para uno o más campos.

Después de agregar un proveedor de sugerencias, puede usar los puntos de conexión `suggestion` y `autocomplete` de API REST o los métodos `DocumentsOperationsExtensions.Suggest` y `DocumentsOperationsExtensions.Autocomplete` de .NET para enviar un término de búsqueda parcial y recuperar una lista de resultados sugeridos o términos autocompletados que se mostrarán en la interfaz de usuario.

Priorización de resultados y puntuación personalizada

De forma predeterminada, los resultados de la búsqueda se ordenan por una puntuación de relevancia, la cual se calcula en función de un algoritmo de mide la frecuencia del término y la frecuencia inversa del documento (TF/IDF). Puede personalizar la forma en que se calcula esta puntuación definiendo un perfil de puntuación que aplica un valor de ponderación a campos específicos; es decir, aumenta la puntuación de búsqueda de los documentos cuando el término de búsqueda se encuentra en esos campos. Además, puede priorizar los resultados en función de los valores de campo; por ejemplo, aumentar la puntuación de relevancia de los documentos según la fecha de su última modificación o su tamaño de archivo.

Después de definir un perfil de puntuación, puede especificar su uso en una búsqueda individual, o bien puede modificar una definición de índice para que use el perfil de puntuación personalizado de forma predeterminada.

Sinónimos

A menudo, se puede hacer referencia a lo mismo de varias maneras. Por ejemplo, alguien que busque información sobre el Reino Unido podría usar cualquiera de los siguientes términos:

Reino Unido

Reino Unido

Gran Bretaña*

GB*

*Para ser precisos, Reino Unido y Gran Bretaña son entidades diferentes, pero se suelen confundir; por tanto, es razonable asumir que alguien que busque "Reino Unido" puede estar interesado en resultados que hagan referencia a "Gran Bretaña".

Para ayudar a los usuarios a encontrar la información que necesitan, puede definir mapas de sinónimos que vinculen términos relacionados. Después, puede aplicar esos mapas de sinónimos a campos individuales de un índice, de modo que, cuando un usuario busque un término determinado, los documentos con los campos que contengan dicho término o cualquiera de sus sinónimos se incluirán en los resultados.

Resumen

En este módulo, ha aprendido a usar Búsqueda de Azure AI para crear una solución de búsqueda con inteligencia artificial que consta de lo siguiente:

Un origen de datos donde se almacenan los datos que se van a indexar (aunque también puede enviar datos directamente a un índice mediante la API).

Un conjunto de aptitudes que define y enriquece la canalización de aptitudes cognitivas para enriquecer los datos de índice.

Un índice que define los campos y que el usuario puede consultar.

Un indexador que rellena los campos del índice con los valores extraídos de los datos de origen.

Ahora que ya ha completado este módulo, ya sabe cómo:

Creación de una solución de Búsqueda de Azure AI

Desarrollar una aplicación de búsqueda

Puede usar las API REST de Búsqueda de Azure AI o los SDK para crear y administrar objetos de índice, así como para implementar una aplicación cliente que consulte el índice para recuperar información.

Para más información sobre Búsqueda de Azure AI, consulte la documentación del servicio.