Desarrollo de soluciones de inteligencia artificial generativa con Azure OpenAI Service

Azure OpenAI Service proporciona acceso a los potentes y grandes modelos de lenguaje de OpenAI, como los modelos ChatGPT, GPT, Codex y Embeddings. Estos modelos permiten que varias soluciones de procesamiento de lenguaje natural (NLP) comprendan, comuniquen y generen contenido. Los usuarios pueden acceder al servicio mediante las API REST, los SDK y Azure OpenAI Studio.

Requisitos previos

Antes de iniciar esta ruta de aprendizaje, debe contar con lo siguiente:

Familiaridad con Azure y Azure Portal

Experiencia en programación con C# en Python. Si no tiene ninguna experiencia anterior en programación, se recomienda completar la ruta de aprendizaje Primeros pasos con C# o Primeros pasos con Python antes de empezar con esta.

Introducción

Supongamos que desee compilar una aplicación de soporte técnico que resuma el texto y sugiera código. Para crear esta aplicación, va a usar las funcionalidades que ve en ChatGPT, un bot de chat creado por la empresa de investigación OpenAI que emplea la entrada de lenguaje natural de un usuario y devuelve una respuesta similar a la humana creada por la máquina.

Los modelos generativos de inteligencia artificial potencian la capacidad de ChatGPT para producir nuevos contenidos, como texto, código e imágenes, a partir de instrucciones en lenguaje natural. Muchos modelos generativos de inteligencia artificial son un subconjunto de algoritmos de aprendizaje profundo. Estos algoritmos admiten diversas cargas de trabajo de visión, habla, lenguaje, decisión, búsqueda, etc.

Azure OpenAI Service lleva los modelos generativos de inteligencia artificial a la plataforma de Azure, lo que le permite desarrollar unas soluciones de inteligencia artificial eficaces que sacan partido de la seguridad, escalabilidad e integración de otros servicios proporcionados por la plataforma en la nube de Azure. Estos modelos están disponibles para compilar aplicaciones mediante una API REST, varios SDK y una interfaz de Studio. Este módulo le guía a través de la experiencia de Inteligencia artificial de Azure Studio, lo que le proporciona la base para desarrollar soluciones con inteligencia artificial generativa.

Acceso a Azure OpenAI Service

El primer paso para crear una solución de IA generativa con Azure OpenAI es aprovisionar un recurso de Azure OpenAI en la suscripción de Azure. El servicio Azure OpenAI ya está disponible para todas las cuentas de Azure, con algunas características avanzadas (como filtros de contenido personalizados) restringidas detrás de una directiva de acceso limitado.

Una vez que tenga acceso a Azure OpenAI Service, puede empezar creando un recurso en Azure Portal o con la interfaz de la línea de comandos (CLI) de Azure.

Cree un recurso de Azure OpenAI Service en Azure Portal.

Al crear un recurso de Azure OpenAI Service, debe proporcionar un nombre de suscripción, un nombre de grupo de recursos, una región, un nombre de instancia único y seleccionar un plan de tarifa.

Creación de un recurso de Azure OpenAI Service en la CLI de Azure

Para crear un recurso de Azure OpenAI Service desde la CLI, consulte este ejemplo y reemplace las siguientes variables por las suyas:

MyOpenAIResource: reemplace por un nombre único para el recurso

OAIResourceGroup: reemplace por el nombre del grupo de recursos

eastus: reemplace por la región para implementar el recurso

subscriptionID: reemplace por su id. de suscripción

CLI.NET:

```
az cognitiveservices account create \
-n MyOpenAIResource \
-g OAIResourceGroup \
-l eastus \
--kind OpenAI \
--sku s0 \
--subscription subscriptionID
```

Disponibilidad regional

Azure OpenAI Service proporciona acceso a varios tipos de modelos. Algunos modelos solo están disponibles en regiones específicas. Consulte la guía de disponibilidad de modelos de Azure OpenAI para consultar la disponibilidad regional. Puede crear dos recursos de Azure OpenAI por región.

Uso de Inteligencia artificial de Azure Studio

Inteligencia artificial de Azure Studio proporciona acceso a los recursos de administración de modelos, implementación, experimentación, personalización y aprendizaje.

Puede acceder a Inteligencia artificial de Azure Studio desde Azure Portal después de crear un recurso o en https://ai.azure.com/ al iniciar sesión en la cuenta de Azure. Durante el flujo de trabajo de inicio de sesión, seleccione el directorio, la suscripción de Azure y el recurso de Azure OpenAI adecuados.

Cuando abra Inteligencia artificial de Azure Studio por primera vez, querrá ir a la página Azure OpenAI, seleccionar el recurso si aún no lo ha hecho e implementar el primer modelo. Para ello, seleccione la página Implementaciones, desde donde puede implementar un modelo base y empezar a experimentar con él.

Exploración de tipos de modelos de inteligencia artificial generativa

Para empezar a compilar con Azure OpenAI, debe elegir un modelo base e implementarlo. Microsoft proporciona modelos base y la opción para crear modelos base personalizados. Este módulo abarca los modelos básicos disponibles actualmente.

Azure OpenAI incluye varios tipos de modelo:

Los modelos GPT-4 son la última generación de modelos generativos previamente entrenados (GPT) que pueden generar finalizaciones de código y lenguaje natural basadas en mensajes de lenguaje natural.

Los modelos GPT 3.5 pueden generar finalizaciones de código y lenguaje natural basadas en mensajes de lenguaje natural. En concreto, los modelos GPT-35-turbo están optimizados para interacciones basadas en chat y funcionan bien en la mayoría de los escenarios de IA generativos.

Los modelos de incrustaciones convierten texto en vectores numéricos y son útiles en escenarios de análisis de lenguaje, como comparar orígenes de texto con similitudes.

Los modelos DALL-E se usan para generar imágenes basándose en mensajes de lenguaje natural. Actualmente, los modelos DALL-E están en versión preliminar. Los modelos DALL-E no aparecen en la interfaz de Inteligencia artificial de Azure Studio y no es necesario implementarlos explícitamente.

Los modelos difieren según la velocidad, el costo y la forma en que realizan tareas específicas. Puede obtener más información sobre las diferencias y los modelos que se ofrecen más recientes en la documentación de Azure OpenAI Service.

En Inteligencia artificial de Azure Studio, la página Catálogo de modelo enumera los modelos base disponibles y proporciona una opción para crear modelos personalizados adicionales mediante la optimización de los modelos base. Los modelos que tienen un estado Correcto significan que se han entrenado correctamente y se pueden seleccionar para la implementación.

Observará que hay varios modelos más allá de OpenAI disponibles en el Catálogo de modelo, incluidos modelos de Microsoft, Meta, Mistral, etc. Inteligencia artificial de Azure Studio le permite implementar cualquiera de estos modelos para su caso de uso. Este módulo se centrará en los modelos de Azure OpenAI.

Implementación de los modelos de inteligencia artificial generativa

En primer lugar, debe implementar un modelo para chatear o realizar llamadas API para recibir respuestas a mensajes. Al crear una nueva implementación, debe indicar qué modelo base se va a implementar. Puede implementar cualquier número de implementaciones en uno o varios recursos de Azure OpenAI siempre que sus tokens por minuto (TPM) permanezcan dentro de la cuota de implementación.

Implementación mediante Inteligencia artificial de Azure Studio

En la página Implementaciones de Inteligencia artificial de Azure Studio, puede crear una nueva implementación seleccionando un nombre de modelo en el menú. Los modelos base disponibles proceden de la lista de la página de modelos.

En la página Implementaciones de Studio, también puede ver información sobre todas las implementaciones, como el nombre de implementación, el nombre del modelo, la versión del modelo, el estado, la fecha de creación, etc.

Implementación con la CLI de Azure

Asimismo puede implementar un modelo mediante la consola. Con este ejemplo, reemplace las siguientes variables por sus propios valores de recurso:

OAIResourceGroup: reemplace por el nombre del grupo de recursos

MyOpenAIResource: reemplácelo por el nombre del recurso.

MyModel: reemplácelo por un nombre único para su modelo.

gpt-35-turbo: reemplácelo por el modelo base que desee implementar.

CLI de .NET:

```
az cognitiveservices account deployment create \
  -g OAIResourceGroup \
  -n MyOpenAIResource \
  --deployment-name MyModel \
  --model-name gpt-35-turbo \
  --model-version "0301"  \
  --model-format OpenAI \
  --sku-name "Standard" \
  --sku-capacity 1
```

Implementación mediante la API de REST

Puede implementar un modelo mediante la API REST. En el cuerpo de la solicitud, especifique el modelo base que desee implementar. Consulte un ejemplo en la documentación de Azure OpenAI.

Uso de mensajes para obtener finalizaciones de los modelos

Una vez implementado el modelo, puede probar cómo se finalizan los mensajes. Un mensaje es la parte de texto de una solicitud que se envía al punto de conexión de las finalizaciones del modelo implementado. Las respuestas se conocen como finalizaciones, que pueden aparecer en forma de texto, código o en otros formatos.

Tipos de avisos

Los mensajes se pueden agrupar en tipos de solicitudes según la tarea.

Tipos de avisos

Tipo de tarea: Contenido de clasificación

Ejemplo de mensaje: Tweet: Disfruté del viaje.

Ejemplo de finalización: Opinión: Positivo

Tipo de tarea: Generación de contenido nuevo

Ejemplo de mensaje: Lista de formas de viajar

Ejemplo de finalización: 1. Bicicleta 2. Coche...

Tipo de tarea: Conversación

Ejemplo de mensaje: Asistente de IA agradable

Ejemplo de finalización: Ver ejemplos

Tipo de tarea: Transformación (traducción y conversión de símbolos)

Ejemplo de mensaje: Inglés: Hola

Ejemplo de finalización: Francés: bonjour

Tipo de tarea: Resumen del contenido

Ejemplo de mensaje: Se proporciona un resumen del contenido {text}

Ejemplo de finalización: El contenido comparte métodos de aprendizaje automático.

Tipo de tarea: Continuar desde donde lo dejó

Ejemplo de mensaje: Una manera de cultivar tomates es plantar semillas.

Ejemplo de finalización: Dar respuestas con hechos

Tipo de tarea: Dar respuestas con hechos

Ejemplo de mensaje: ¿Cuántas lunas tiene la Tierra?

Ejemplo de finalización: Uno

Calidad de la finalización

La calidad de las finalizaciones que obtendrá de una solución de IA generativa depende de varios factores.

La forma en que se diseñan los mensajes. Más información sobre la ingeniería de mensajería aquí.

Parámetros del modelo (se tratan a continuación)

Los datos con los que se entrena el modelo, que se pueden adaptar mediante la optimización del modelo con personalización.

El entrenamiento de un modelo personalizado supone más control sobre las finalizaciones devueltas que la ingeniería de mensajería y el ajuste de los parámetros.

Realización de llamadas

Puede empezar a realizar llamadas al modelo implementado a través de la API REST, Python, C# o desde Studio. Si el modelo implementado tiene una base de modelo GPT-3.5 o GPT-4, use la documentación de finalizaciones de chat, con los diferentes puntos de conexión de solicitud y variables necesarios que para otros modelos base.

Prueba de modelos en el área de juegos de Inteligencia artificial de Azure Studio

Las áreas de juegos son interfaces útiles de Inteligencia artificial de Azure Studio que puede usar para experimentar con los modelos implementados sin necesidad de desarrollar una aplicación cliente propia. Inteligencia artificial de Azure Studio ofrece varias áreas de juegos con diferentes opciones de ajuste de parámetros.

Área de juegos de finalizaciones

Esta área de juegos permite realizar llamadas a los modelos implementados mediante una interfaz de entrada y salida de texto y ajustar los parámetros. Debe seleccionar el nombre de implementación del modelo en "Implementaciones". Opcionalmente, puede usar los ejemplos proporcionados para empezar y, luego, puede escribir sus propios mensajes.

Parámetros del área de juegos de finalizaciones

Hay muchos parámetros que puede ajustar para cambiar el rendimiento del modelo:

Temperatura: controla la aleatoriedad. Reducir la temperatura significa que el modelo genera respuestas más repetitivas y deterministas. Aumentar la temperatura da como resultado respuestas más inesperadas o creativas. Intente ajustar la temperatura o el Top P pero no ambos.

Longitud máxima (tokens): establezca un límite en el número de tokens por respuesta del modelo. La API admite hasta 4000 tokens compartidos entre la solicitud (incluidos el mensaje del sistema, los ejemplos, el historial de mensajes y la consulta del usuario) y la respuesta del modelo. Un token equivale aproximadamente a cuatro caracteres de un texto típico en inglés.

Secuencias de detención: haga que las respuestas se detengan en un punto deseado, como al final de una oración o una lista. Especifique hasta cuatro secuencias en las que el modelo dejará de generar más tokens en una respuesta. El texto devuelto no contendrá la secuencia de detención.

Probabilidades principales: de forma similar a la temperatura, este parámetro controla la aleatoriedad, pero usa un método diferente. Al reducir Top P, la selección de tokens del modelo se reduce a los más probables. Aumentar Top P permite al modelo elegir entre tokens con alta y baja probabilidad. Intente ajustar la temperatura o el Top P pero no ambos.

Penalización de frecuencia: reduzca la posibilidad de repetir un token proporcionalmente en función de la frecuencia con la que ha aparecido en el texto hasta ahora. Así se reduce la probabilidad de repetir exactamente el mismo texto en una respuesta.

Penalización de presencia: reduzca la posibilidad de repetir cualquier token que haya aparecido en el texto hasta ahora. Así aumenta la probabilidad de introducir nuevos temas en una respuesta.

Texto anterior a la respuesta: inserte texto después de la entrada del usuario y antes de la respuesta del modelo. Esto puede ayudar a preparar el modelo para una respuesta.

Texto posterior a la respuesta: inserte texto después de la respuesta generada por el modelo para animar al usuario a realizar más aportaciones, como cuando se modela una conversación.

Área de juegos de chat

El área de juegos de chat se basa en una interfaz de entrada de conversación y salida de mensaje. Puede inicializar la sesión con un mensaje del sistema para configurar el contexto de chat.

En el área de juegos de chat, puede agregar algunos ejemplos. El término "algunos ejemplos" hace referencia a proporcionar ejemplos para ayudar al modelo a aprender lo que necesita hacer. Puede considerarse lo contrario de "sin ejemplos", que no proporciona ningún ejemplo.

En la configuración del Asistente, puede proporcionar algunos ejemplos de lo que puede ser la entrada de usuario y cuál debe ser la respuesta del asistente. El asistente intenta imitar las respuestas que se incluyen aquí en cuanto al tono, las reglas y el formato que ha definido en el mensaje del sistema.

Parámetros del área de juegos de chat

El área de juegos Chat, como el área de juegos Finalizaciones, también incluye parámetros para personalizar el comportamiento del modelo. El área de juegos Chat también admite otros parámetros que no están disponibles en el área de juegos Finalizaciones. Entre ellas se incluyen las siguientes:

Respuesta máxima: establezca un límite en el número de tokens por respuesta del modelo. La API admite hasta 4000 tokens compartidos entre la solicitud (incluidos el mensaje del sistema, los ejemplos, el historial de mensajes y la consulta del usuario) y la respuesta del modelo. Un token equivale aproximadamente a cuatro caracteres de un texto típico en inglés.

Mensajes anteriores incluidos: seleccione el número de mensajes anteriores que se incluirán en cada nueva solicitud de API. Incluir mensajes anteriores ayuda a proporcionar contexto al modelo para las nuevas consultas de los usuarios. Establecer este número en 10 da como resultado cinco consultas del usuario y cinco respuestas del sistema.

El recuento de tokens actual se puede ver en el área de juegos de chat. Dado que las llamadas API tienen un precio por token y es posible establecer un límite máximo de tokens de respuesta, querrá vigilar el recuento de tokens actual para asegurarse de que la conversación no supere el número máximo.

Resumen

En este módulo se tratan los conceptos básicos de introducción a Azure OpenAI Service centrados en el uso de la Inteligencia artificial de Azure Studio.

Ha aprendido a:

Cree un recurso de Azure OpenAI Service y comprenda los tipos de modelos base de Azure OpenAI.

Use la Inteligencia artificial de Azure Studio, la CLI de Azure y la API REST para implementar un modelo base.

Genere finalizaciones para los mensajes.

Pruebe los modelos en el área de juegos de Studio y empiece a administrar los parámetros del modelo.

Una manera de obtener más información es explorar la documentación de Azure OpenAI Service.