Workshop AI (day 3) Microsoft-AGIC-UniPadova

Le API di Azure OpenAI

Azure mette a disposizione tre principali tipologie di API per interagire con i modelli linguistici.

- La **Completion API** consente di invocare modelli di tipo chat in modalità stateless, perfetta per richieste singole senza contesto.
- La **Batch API** è invece progettata per elaborare grandi volumi di dati in parallelo, adatta a scenari come il pre-processing di documenti su larga scala.
- Infine, la **Assistants API** permette di creare agenti conversazionali stateful, arricchiti da configurazioni, conoscenze e azioni personalizzate. È su questa API che si basa l'intero concetto di agenti descritto in seguito.

Gli Agenti di Azure Agent Service

Quando si parla di agenti, ci si riferisce a una funzionalità basata sull'**Assistants API**, progettata per estendere i classici modelli di tipo chat con uno strato di **gestione dello stato**. A differenza della chat completion, dove ogni chiamata è stateless e la conversazione viene dimenticata una volta terminata, con gli agenti è possibile mantenere la memoria del contesto. L'agente conserva i messaggi precedenti (system, user, assistant) ma anche le configurazioni iniziali che gli vengono fornite, come un nome, delle impostazioni personalizzate, un collegamento a una base di conoscenza, ecc.

La presenza dello stato è ciò che distingue un agent da una semplice chat. L'agente rimane **persistente** finché non viene esplicitamente eliminato e fino ad allora è capace di mantenere uno stato coerente durante più interazioni.

Dal punto di vista della conoscenza, un agente può accedere a due fonti principali.

- La prima è una modalità **RAG (Retrieval-Augmented Generation)**, cioè la capacità di recuperare informazioni in tempo reale da un set di dati fornito dall'utente.
- La seconda è la **Bing Search**, che permette di estendere la conoscenza del modello al web. Quest'ultima è comoda ma comporta dei rischi in termini di accuratezza, motivo per cui è bene valutarla con attenzione.

Un concetto fondamentale in questo ecosistema è quello di **thread**. Ogni thread rappresenta una conversazione specifica. Ogni volta che si apre e chiude una sessione, ad esempio nel playground, si sta creando e concludendo un thread. È grazie a questo meccanismo che l'agente riesce a "ricordare" il contesto della conversazione.

Differenze tra Agent Service e Azure AI Search

L'**Agent Service** è una soluzione molto più flessibile rispetto ad Azure AI Search. È pensato per scenari con una **quantità limitata di documenti** (ad oggi fino a 10.000) e non richiede una strutturazione complessa del dato. Non è necessario definire uno schema, creare un index, o configurare uno skillset con skill specifiche. L'agente **può funzionare "out of the box**", offrendo un comportamento intelligente anche con una configurazione minima.

Al contrario, **Azure AI Search** richiede una progettazione più strutturata. I dati devono essere organizzati all'interno di un indice, e l'enrichment semantico si realizza attraverso skill definite in uno skillset. Questo approccio più rigido offre però una **grande scalabilità**, rendendolo adatto a scenari in cui è necessario gestire numerosi documenti. La configurazione iniziale è certamente più onerosa rispetto a un agent, ma il risultato è una piattaforma altamente robusta ed efficiente.

In generale, è fondamentale scegliere l'architettura in base alla natura dei dati e allo use case. Questo non significa selezionare un solo servizio, ma costruire un'architettura in cui più servizi lavorano insieme in modo sinergico. La combinazione corretta di componenti permette di raggiungere la massima efficienza possibile, sia in termini di prestazioni che di costi.