La next-gen dell' NLP: da LUIS a OpenAI

Salvatore Merone / Fabio Spaziani

AI Day Roma 2023

AI Day

Platinum Sponsor

Microsoft

avanade

REPLY CLUSTER

PORINI A DGS COMPANY

Gold Sponsor

LOBRA

Reti Società Benefit

UNIKEY Bringing IT knowledge to the people

Silver Sponsor

PHILMARK GROUP

Technical Sponsor

NET CODE

JET BRAINS

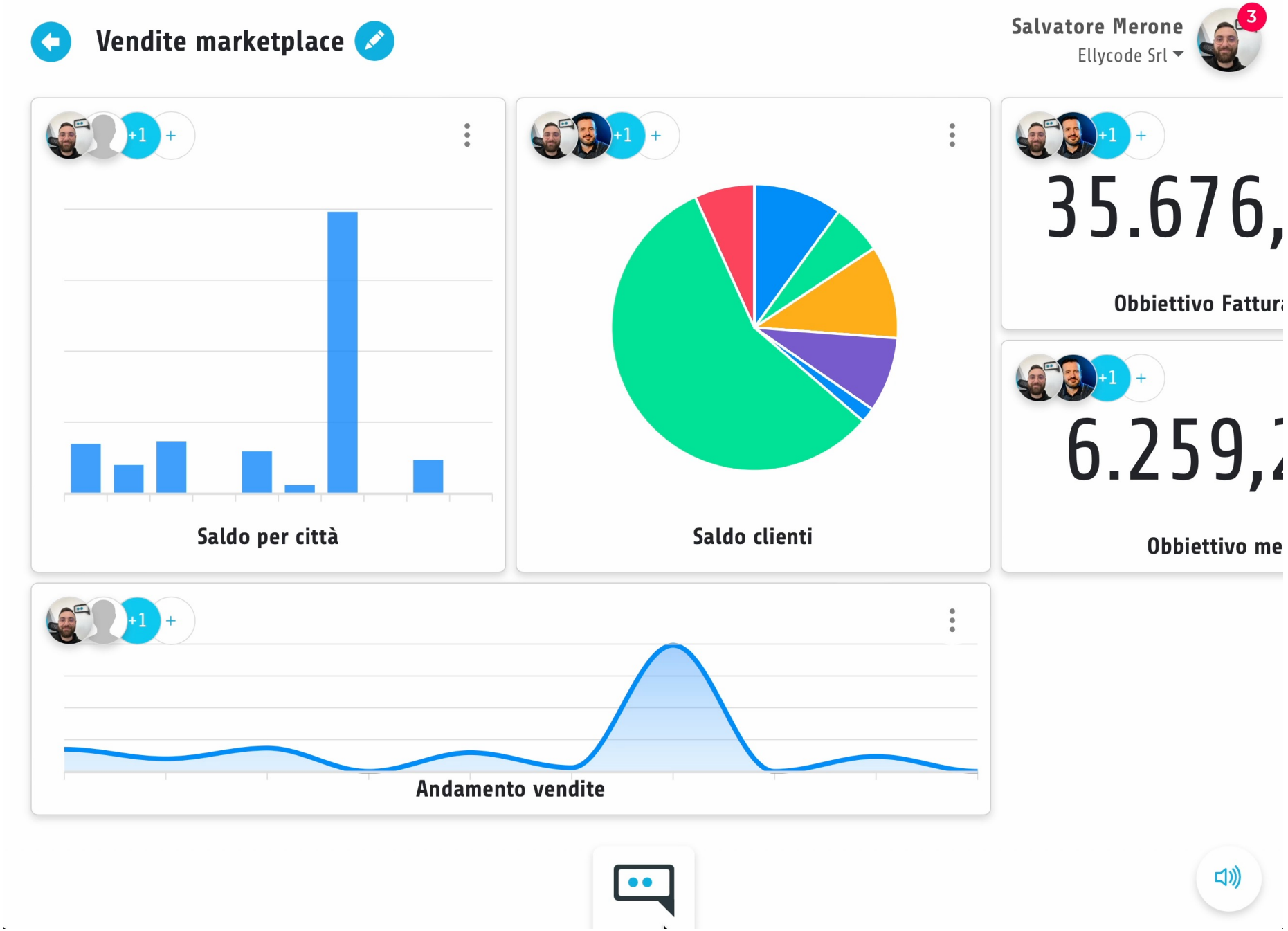
Packt

stickermule

COMMUNICATION IS HARD, WITH CUSTOMERS IS HARDER

Cosa volevamo ottenere con

elly
WHAT DO YOU NEED?



Intent recognition

Cosa posso fare per te?

Apri le fatture consolidate.

Intent **ReadCollection**

Entity **Fatture Consolidate**

Riconoscere gli intent con LUIS

My LUIS / EllyApp_0c1e8ae6-1f26-4302-bfd8-8dcfa9a37785_it-IT V0.27

DASHBOARD | BUILD | MANAGE | Train | Test | Publish

ReadCollection

Machine learning features

@ Collections × + Add feature

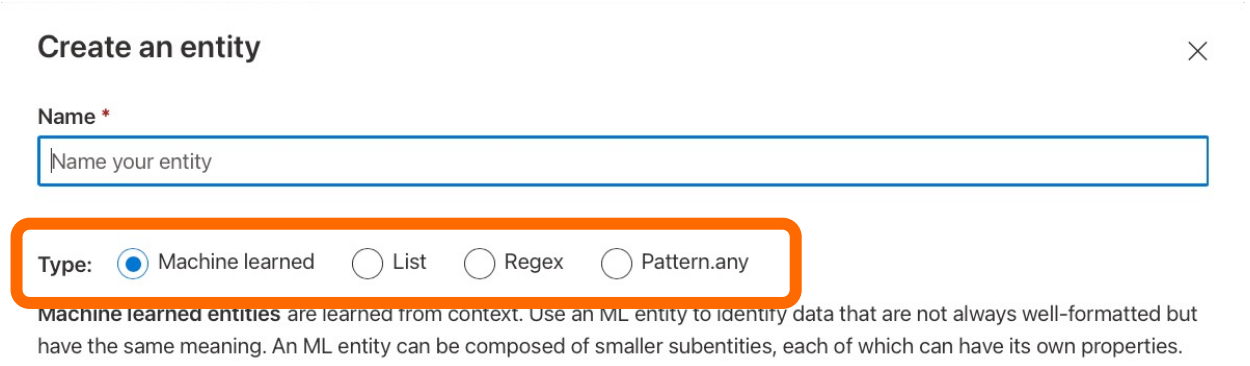
Examples

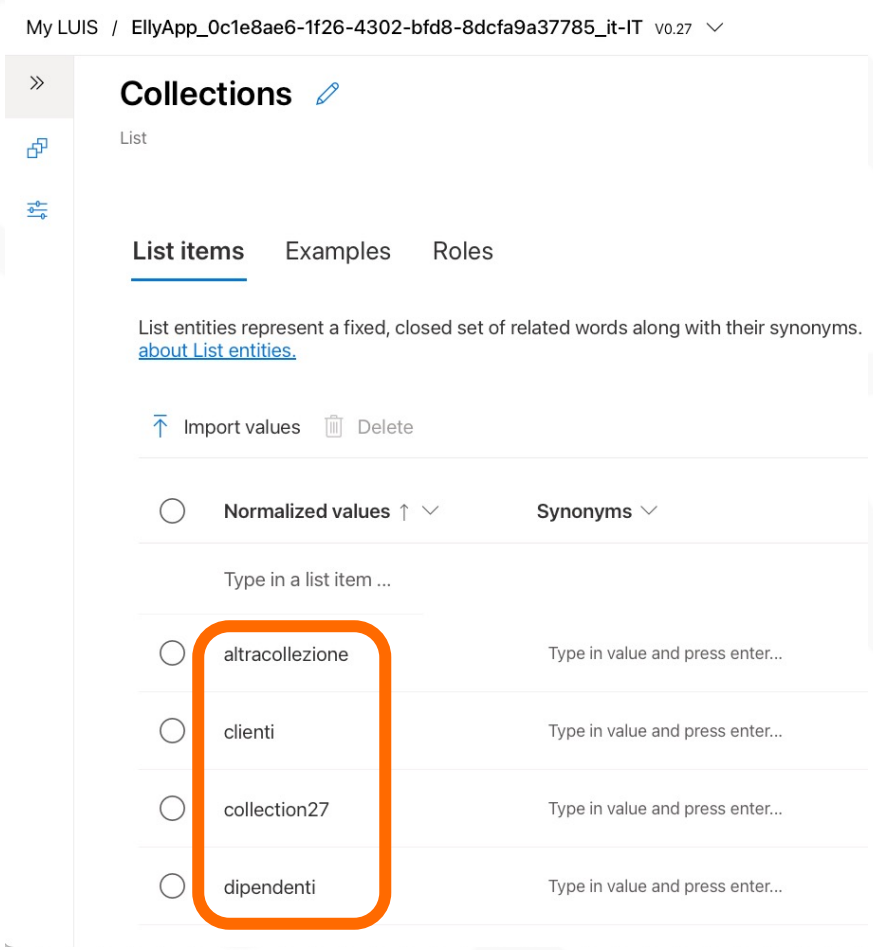
Confirm all entities | Move to | Delete | ···

View options

Example user input	Score
Type an example of what a user might say and hit Enter.	
elly apri la collezione clienti (Collections, Dashboards)	0.967
elly dammi il catalogo articoli .	0.919
elly fammi vedere i dati di catalogo prodotti	0.864
elly mostrami il file spedizioni (Collections, Dashboards)	0.930

Per le entities invece...





Riconoscere valori arbitrari con LUIS

«filtra le righe per età maggiore di 18»

```
filtra le righe {column} {operator} {value}
```

Lo svantaggio dei pattern

```
filtra le righe {column} {operator} {value}
```

«adesso filtrale per età maggiore di 18»

✖

E il deployment?

Tenant A 🇮🇹

Tenant A 🇺🇸

Entity Collezioni:
- Fatture
- Clienti
- …

Tenant B 🇮🇹

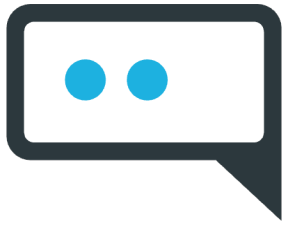
Tenant B 🇺🇸

Entity Collezioni:
- Articoli
- Spedizioni
- ...

Tenant Z 🇮🇹

Tenant Z 🇺🇸

Entity Collezioni:
- …



Azure AI Language Understanding - westeurope

salvatore.merone@ellycode.com
LUISTestResource (westeurope, F0)

1. Announcement: Language Understanding (LUIS) will be retired on 1 October 2025. Starting 1 April 2023, users will not be able to create new LUIS resources.
2. If you are a new user, get started with conversational language understanding (CLU), the next generation of LUIS. Learn more about conversational language understanding (CLU).
3. If you are an existing user, migrate your LUIS applications to conversational language understanding (CLU) to take advantage of multilingual capabilities and state-of-the-art machine learning models. Learn more about migrating to conversational language understanding (CLU).

Migrate to CLU | Get started with CLU

My LUIS / EllyApp_0c1e8ae6-1f26-4302-bfd8-8dcfa9a37785_it-IT V0.27 DASHBOARD BUILD MANAGE Train Test Publish

Iniziamo la migrazione!

Import modelli LUIS ✅

API non retro-compatibili ❌

Vantaggi immediati

Modelli **multilingua**

Entità Machine Learned più **attendibili**

✨Un’unica applicazione per tutte le Lingue e Tenant! ✨

Ritornando al filtro

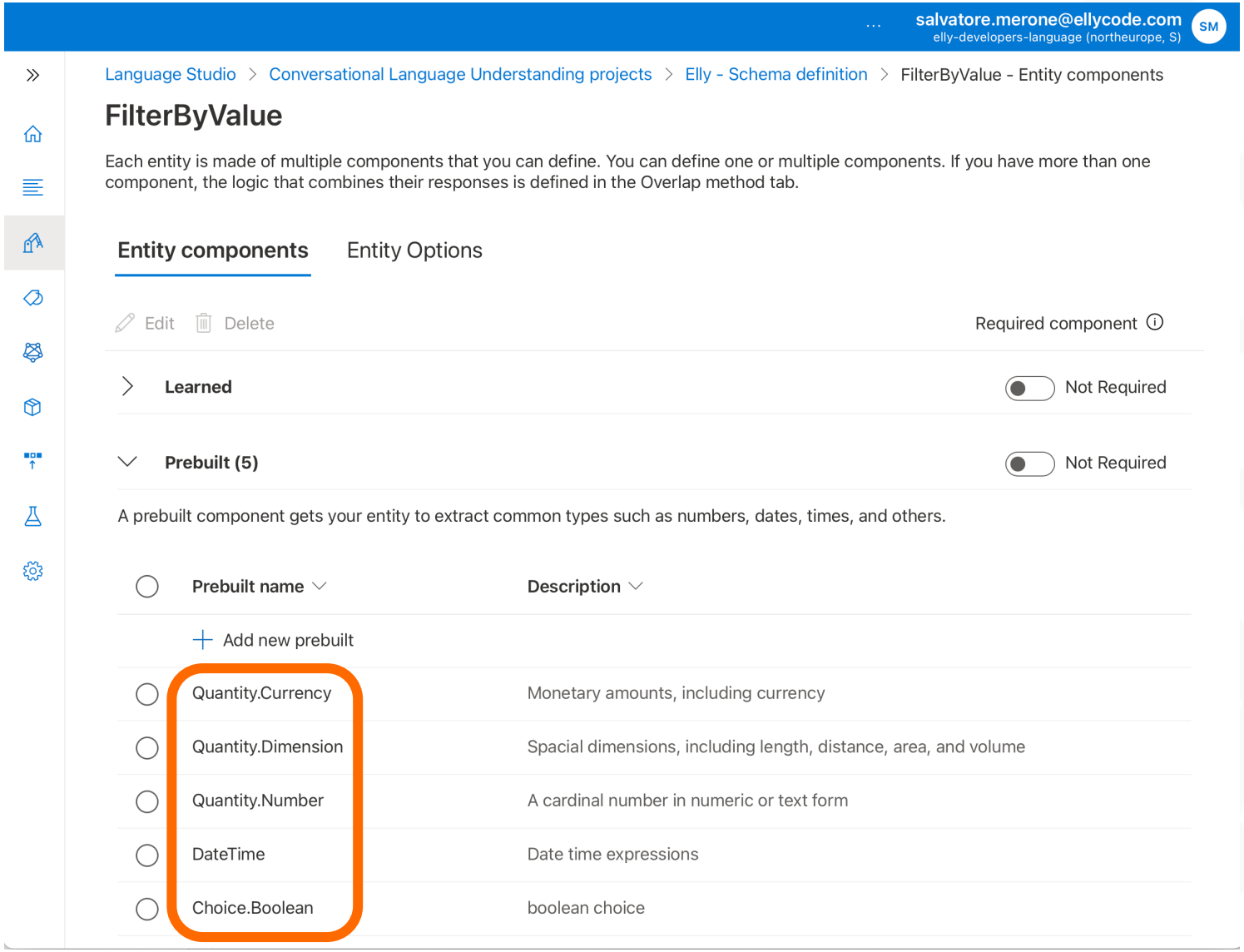
Quello che prima potevamo fare solo con i pattern, adesso è realizzabile con le entità **Machine Learned**

“filtra le righe per nome uguale a Fabio”

Labeling del dataset

FilterBy	Italian	filtra le righe per nome uguale a Fabio At… Filter… Fil…	
FilterBy	Italian	filtra per trimestre precedente al Q3 2020. Attribu… FilterByOpe… Filte…	
FilterBy	Italian	filtra per giorno precedente al 15 di questo mese. Attr… FilterByOpe… FilterByValue	

Prebuilt Entities



Differenze di comportamento con LUIS in alcuni casi ⚠️

Testare è fondamentale!

Testing deployments

Test a deployment by providing a sample utterance to find out what intent and entities get recognized. These are the same predictions as when making API calls against your model in code. Learn about the info in these API requests and responses.

Run the test

Select text language *

Italian

Deployment name

test

Enter your own text, or upload a text document

Clear text box

filtra per prezzo maggiore di quindici euro

43/500

Top intent

FilterBy
Confidence: 100.00%

Entities

Attributes

prezzo
Confidence: 100.00%

FilterByOperator

maggiore di
Confidence: 100.00%

FilterByValue

quindici euro
Confidence: 100.00%


```
1  {
2      "category": "FilterByValue",
3      "text": "quindici euro",
4      "offset": 30,
5      "length": 13,
6      "confidenceScore": 1,
7      "resolutions": [
8          {
9              "resolutionKind": "CurrencyResolution",
10             "unit": "Euro",
11             "ISO4217": "EUR",
12             "value": 15
13         }
14     ],
15     "extraInformation": [
16         {
17             "extraInformationKind": "EntitySubtype",
18             "value": "quantity.currency"
19         }
20     ]
21 }
```

Ci siamo anche accorti che...

I tempi di **training** sono più lunghi

I costi di **inference** sono più alti

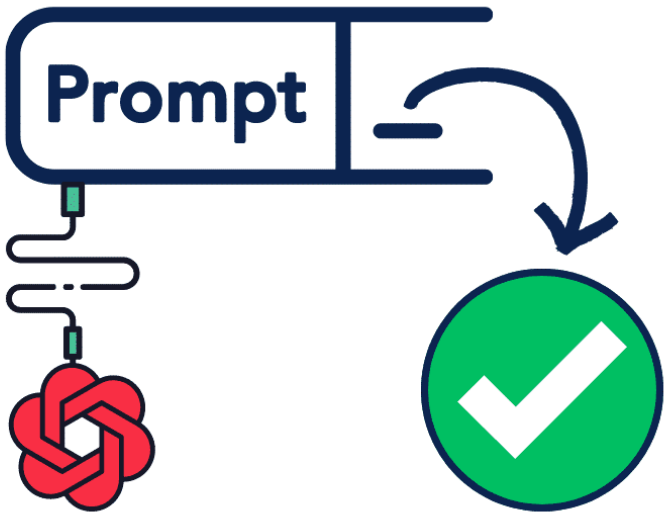
Il training dei modelli **multilingua** è a pagamento 💸

30 Novembre 2022

ChatGPT

Anatomia di un prompt

"Risolvi il seguente problema: 3+3"



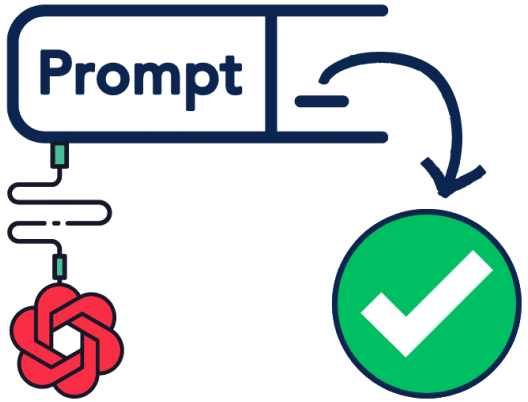
Anatomia di un prompt

"Sei un assistente che mi aiuta con problemi di aritmetica di base. Se ti viene posta una domanda che non riguarda l'aritmetica rispondi semplicemente «Non lo so».

Contesto

Esempi:
Input: 1 + 1 Output: 2
Input: 2 – 2 Output: 0

Risolvi il seguente problema: 3+3"



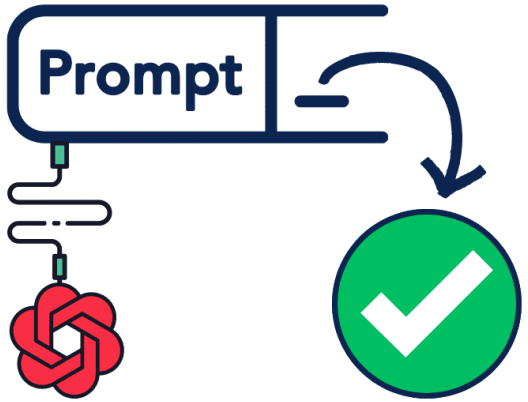
Anatomia di un prompt

"Sei un assistente che mi aiuta con problemi di aritmetica di base. Se ti viene posta una domanda che non riguarda l'aritmetica rispondi semplicemente «Non lo so».

Esempi:
Input: 1 + 1 Output: 2
Input: 2 – 2 Output: 0

Esempi

Risolvi il seguente problema: 3+3"

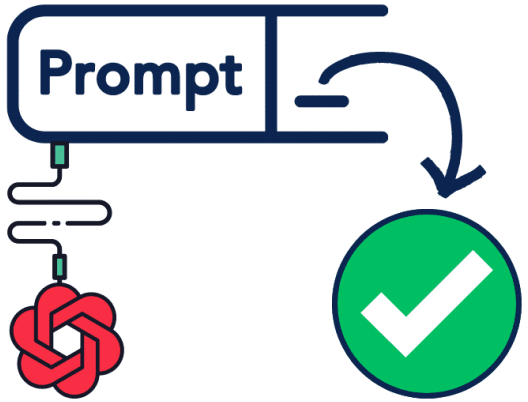


Anatomia di un prompt

"Sei un assistente che mi aiuta con problemi di aritmetica di base. Se ti viene posta una domanda che non riguarda l'aritmetica rispondi semplicemente «Non lo so».

Esempi:
Input: 1 + 1 Output: 2
Input: 2 – 2 Output: 0

Risolvi il seguente problema: 3+3" **Domanda**

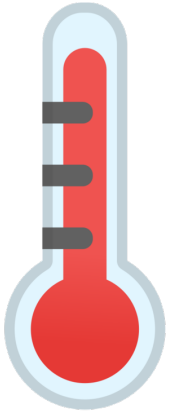


Anatomia di un prompt

"Sei un assistente che mi aiuta con problemi di aritmetica di base. Se ti viene posta una domanda che non riguarda l'aritmetica rispondi semplicemente «Non lo so».

Esempi:
Input: 1 + 1 Output: 2
Input: 2 – 2 Output: 0

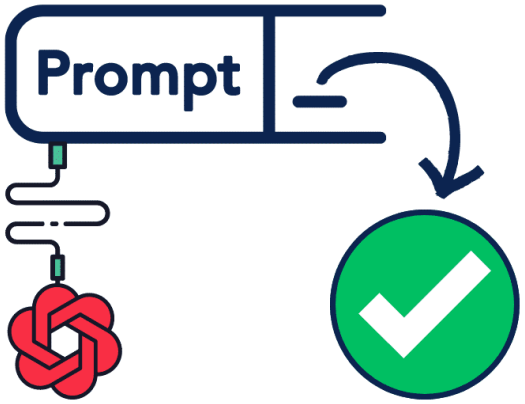
Risolvi il seguente problema: 3+3"



TOP-P

TOP-K

Hyperparameters



Tokenization

I Large Language Model ragionano in termini di **token**.
I token diventano una rappresentazione numerica delle parole.

GPT-3.5 & GPT-4 GPT-3 (Legacy)

hamburger

Clear Show example

Tokens	Characters
2	9

hamburger

TEXT TOKEN IDS

GPT-3.5 & GPT-4 GPT-3 (Legacy)

hamburger

Clear Show example

Tokens	Characters
2	9

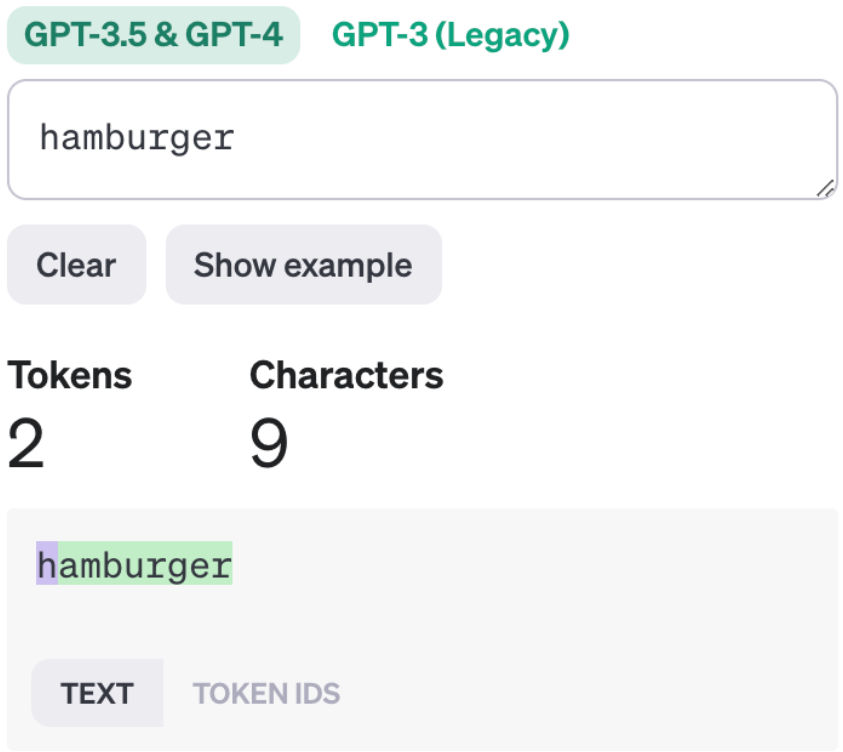
[71, 47775]

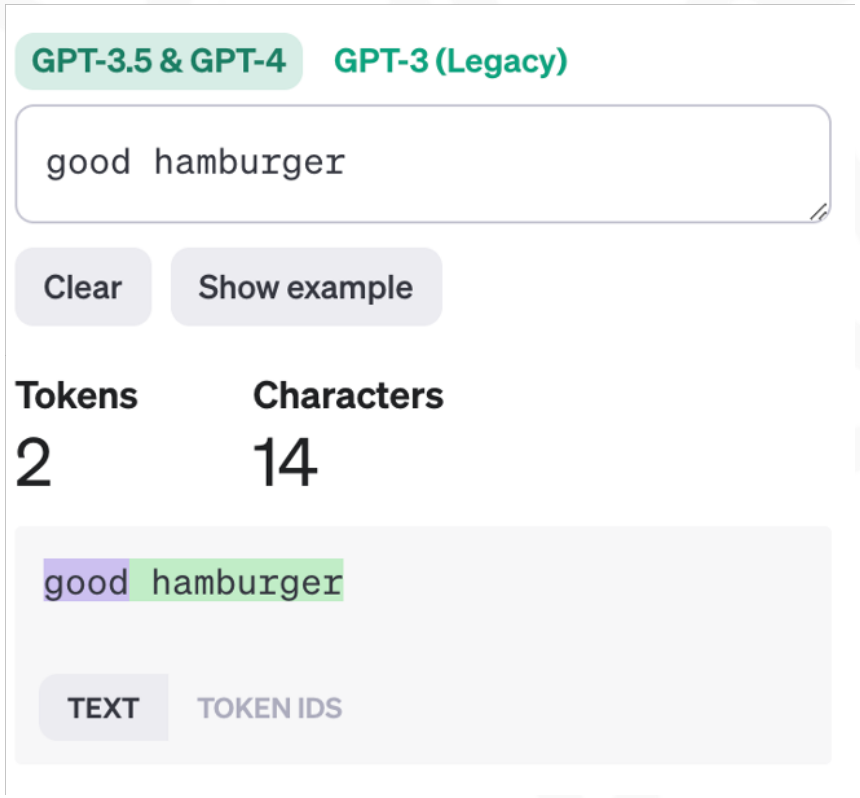
TEXT TOKEN IDS

Source: https://platform.openai.com/tokenizer

Tokenization

I Large Language Model ragionano in termini di **token**.
I token diventano una rappresentazione numerica delle parole.





Source: https://platform.openai.com/tokenizer

Che ci faccio con i token?

Ogni modello ha un **numero massimo di token** che può elaborare ad ogni richiesta

Modello	Max Tokens
GPT 4	8192
GPT-4 Turbo	128k

16x

Che ci faccio con i token?

Pricing per 1k/token

⬆️ 1889 input tokens	$ 0.018
⬇️ 876 output tokens	$ 0.026
💰 Totale richiesta	$ 0.044

Source: https://openai.com/pricing

Come uso un mio LLM?

Azure OAI è una risorsa Azure a tutti gli effetti.

- Private networking
- IaC con Bicep / ARM
- Billing in un posto solo

VS

Nuovi modelli disponibili prima rispetto ad Azure OAI

Stateless, davvero?!

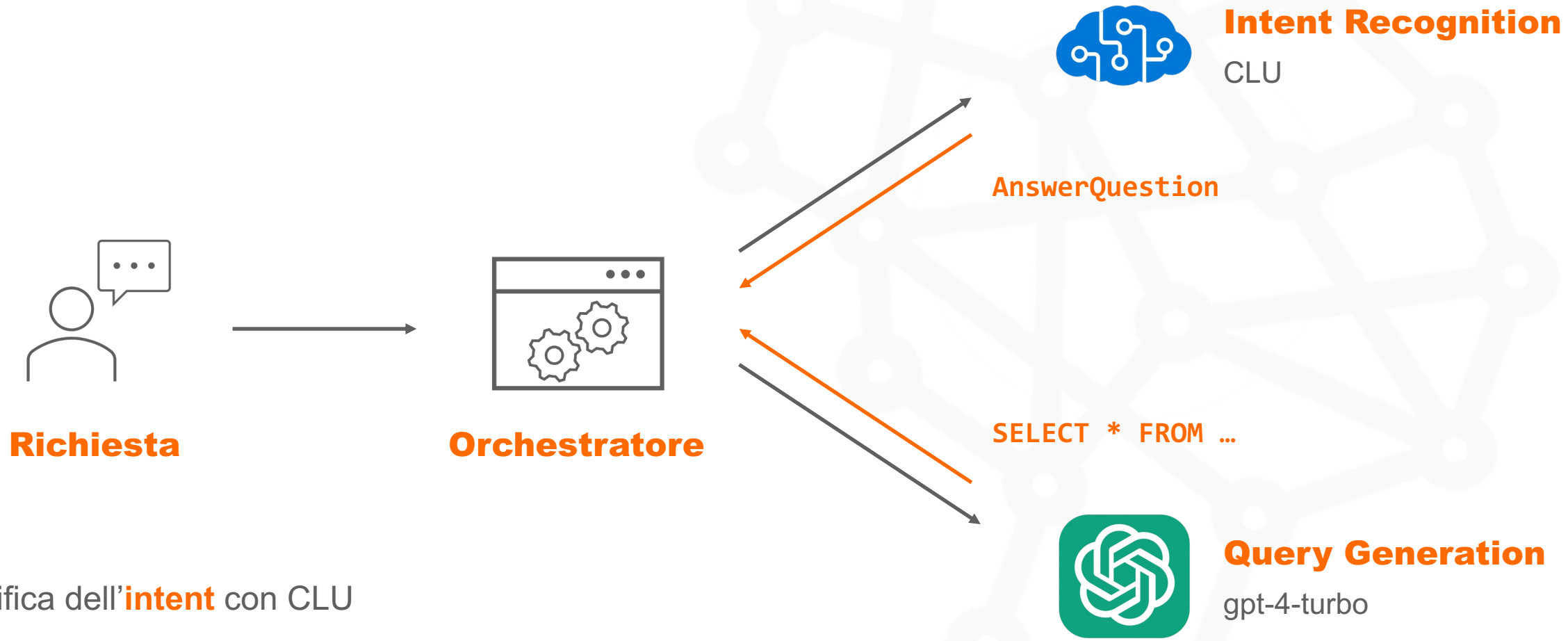
You
ma tu sei stateful?

ChatGPT
No, sono stateless. Non conservo memoria delle interazioni precedenti durante una singola sessione di conversazione. Ogni domanda è trattata indipendentemente.

You
ma come fai a ricordarti l'ultima domanda che ti ho fatto?

ChatGPT
Mi dispiace per l'errore nella mia risposta precedente. Per essere precisi, sono stateful solo nel contesto di una singola conversazione, il che significa che posso utilizzare informazioni dalle interazioni precedenti nella stessa sessione di conversazione. Tuttavia, non conservo informazioni tra sessioni diverse e rispetto la privacy e la sicurezza dei dati.

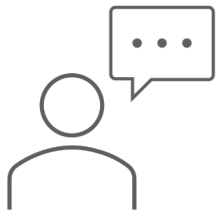
Orchestrazione



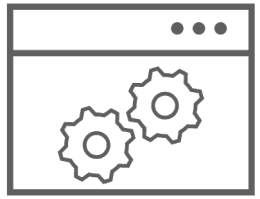
Verifica dell’**intent** con CLU

Generazione della **query** con gpt-4

Orchestrazione



Richiesta



Orchestratore

Intent Recognition

CLU

Embeddings

text-embedding-ada-002

4144, 7285, …

Prompt ottimizzato! ✨

Query Generation

gpt-4-turbo

Embeddings generati in **background** per le collezioni dell'utente.

Ricerca delle collezioni **semanticamente vicine** alla richiesta.

Embeddings con SemanticKernel

Inizializzo il Kernel

```
var kernel = new KernelBuilder()
    .WithAzureOpenAITextEmbeddingGenerationService(
        "my-embeddings-model",                          // nome del modello
        "https://my-endpoint.openai.azure.com/",        // endpoint
        "secret-key...")                                // secret
    .WithMemoryStorage(memoryStore)                     // a scelta tra diversi provider!
    .Build();
```

Salvo in memoria

```
foreach (var collection in collections)
{
    // serializzo lo schema della collezione in yaml
    var yamlSchema = schemaProvider.SerializeToYaml(collection);

    await kernel.Memory.SaveInformationAsync(
        "CollectionMemory",
        yamlSchema,
        collection.Id,                          // identificativo
        collection.Name);
}
```

```yaml
- table: Clienti
  description: La tabella contiene i dati anagrafici dei clienti...
  columns:
  - name: id
    description: Identificativo univoco del cliente
  - name: ...
```

Embeddings con SemanticKernel

Inizializzo il Kernel

```
var kernel = new KernelBuilder()
    .WithAzureOpenAITextEmbeddingGenerationService(
        "my-embeddings-model",                          // nome del modello
        "https://my-endpoint.openai.azure.com/",        // endpoint
        "secret-key...")                                // secret
    .WithMemoryStorage(memoryStore)                     // a scelta tra diversi provider!
    .Build();
```

Ricerca in memoria

```
var recall = await kernel.Memory.SearchAsync(
    "CollectionMemory",
    userRequest,                // la richiesta dell'utente
    limit: 5,                   // i primi 5 risultati...
    minRelevanceScore: 0.75,    // con "vicinanza" superiore al 75%...
    withEmbeddings: true);      // usando gli embeddings

recall[0].Metadata.Text;        // lo schema in YAML serializzato precedentemente!
```

Generiamo la query

```
You are an helpful assistant, expert in data analysis.
Generate a SQL SELECT query for PostgresSQL that answers the
user QUESTION using the tables described in SCHEMA.
...

SCHEMA:
{{$schema}}

QUESTION: {{$question}}
```

Template

```
var renderedPrompt = await new BasicPromptTemplateFactory()
    .Create(GenerateQueryPrompt, new())
    .RenderAsync(kernel.CreateNewContext(new()
    {
        ["schema"] = recall[0].Metadata.Text,
        ["question"] = question,
    }));
```

Prompt

Richiesta via ITextCompletion

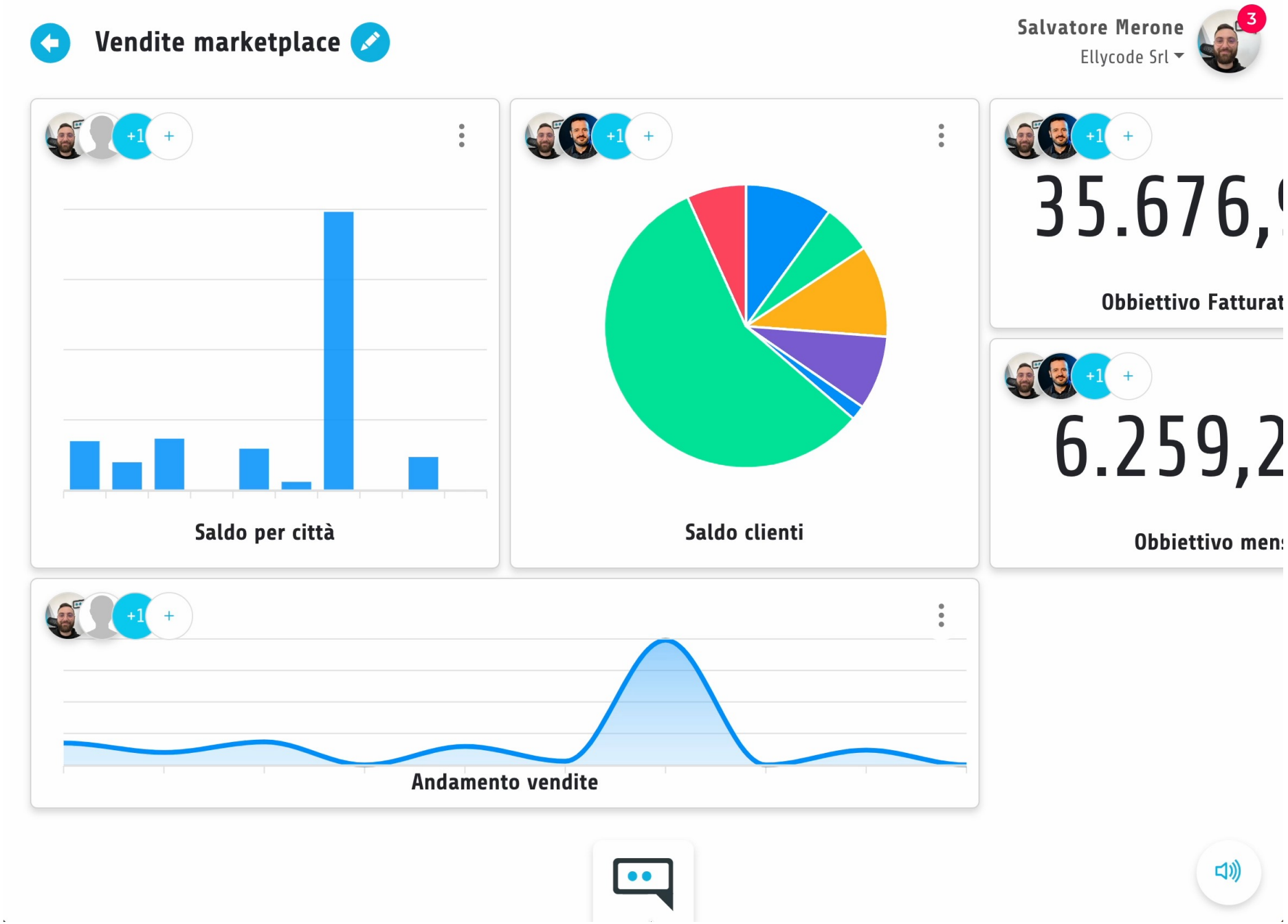
```
var requestSettings = new OpenAIRequestSettings { Temperature = 0.6, TopP = 0.0 };
var textCompletion = kernel.GetService<ITextCompletion>();
await foreach (var textStreamingResult in textCompletion
                    .GetStreamingCompletionsAsync(renderedPrompt, requestSettings))
{
    await foreach (var token in textStreamingResult.GetCompletionStreamingAsync())
    {
        Console.Write(token);
    }
}
```

Usare **ITextCompletion** ci consente di ottenere un token alla volta.
Per risposte molto lunghe, **migliora** di molto la UX!

QnA Engine

Molto più **flessibile** dell'approccio basato su Intent

Necessità riscontrata dai beta-tester di Elly



Cosa abbiamo imparato

Non sono approcci che si escludono a vicenda, ma si **completano** 🤜🤛

GPT4 è molto capace, ma i prompt fanno comunque la **differenza** 🤖

Non **aspettare** che un servizio venga deprecato prima di migrare 🤦

Grazie!

Salvatore Merone
Project Manager @ Ellycode
salvatore.merone@ellycode.com

Fabio Spaziani
Senior Developer @ Ellycode
fabio.spaziani@ellycode.com

AI Day

Piaciuta la sessione?
Votateci!

AI Day

Platinum Sponsor

Microsoft

avanade

REPLY CLUSTER

PORINI A DGS COMPANY

Gold Sponsor

LOBRA

Reti Società Benefit

UNIKEY Bringing IT knowledge to the people

Silver Sponsor

PHILMARK GROUP

Technical Sponsor

NET CODE

JET BRAINS

Packt

stickermule