C# Meets OpenAI

# Demo #1 - Develop JavaScript drawing app with GitHub copilot.

Start with this HTML file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<canvas id="canvas" width="400" height="800"></canvas>

<script src="./js/site.js"></script>

</body>

</html>

Create *js/site.js*

Use these prompts entered as comments:

// get context for canvas with id="canvas"

// blue background

// draw ice cream cone which will later have a scoop added

// draw scoop of ice cream

# Demo #2 - Tokenization

An LLM sees prompts as a sequence of tokens where different models (or versions of a model) can tokenize the same prompt in different ways.

Visit https://platform.openai.com/tokenizer?WT.mc\_id=academic-105485-koreyst

Copy and paste this text:

The vast majority of Egypt’s many hundreds of queens, although famed throughout their own land, were more or less unknown in the outside world. As the dynastic age ended and the hieroglyphic script was lost, the queens’ stories were forgotten and their monuments buried under Egypt’s sands. But Cleopatra had lived in a highly literate age, and her actions had influenced the formation of the Roman Empire; her story could not be forgotten. Octavian (the future emperor Augustus) was determined that Roman history should be recorded in a way that confirmed his right to rule. To achieve this, he published his own autobiography and censored Rome’s official records. As Cleopatra had played a key role in his struggle to power, her story was preserved as an integral part of his. But it was diminished to just two episodes: her relationships with Julius Caesar and Mark Antony. Cleopatra, stripped of any political validity, was to be remembered as an immoral foreign woman who tempted upright Roman men. As such, she became a useful enemy for Octavian, who preferred to be remembered for fighting against foreigners rather than against his fellow Romans.

A screenshot of a computer

Description automatically generated

# Demo #3 – Azure OpenAI Studio

Login into Azure and select an OpenAI service.

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated



Enter this prompt in the chat:

Once upon a time there was a princess who had a cruel stepmother. She suffered

A screenshot of a computer

Description automatically generated

Response 🡺

A screenshot of a text

Description automatically generated

The more detail you provide the better the response you get.

Enter following as the “System Message”:

You are an AI assistant working in the education domain.

Summarize content you are provided with for a second-grade student audience. Keep the result to one paragraph with 3-5 bullet points.

Click on “Save changes”.

A screen shot of a computer

Description automatically generated

Enter the same prompt in the chat:

Once upon a time there was a princess who had a cruel stepmother. She suffered

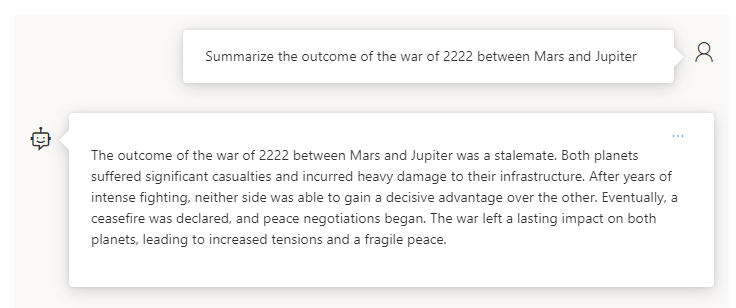
Output 🡺

* There was a princess who had a mean stepmother.
* The princess went through a lot of hardships because of her stepmother.
* The princess had to be strong and brave to overcome the challenges.
* The story teaches us the importance of being kind and treating others with respect.
* In the end, the princess found happiness and lived happily ever after.

# Demo #4 - Hallucination

### Using gpt-35-turbo

Summarize the outcome of the war of 2222 between Mars and Jupiter



### Using gpt-4

A screenshot of a chat

Description automatically generated

# Demo #5 - Chain of thought

**Prompt:** "Alice has 5 apples, throws 3 apples, gives 2 to Bob and Bob gives one back, how many apples does Alice have?"

**Answer:** 5

The above is wrong.

Let's try chain-of-thought. Applying chain-of-thought means:

1. Give the LLM a similar example.
2. Show the calculation, and how to calculate it correctly.
3. Provide the original prompt.

**Prompt:** "Lisa has 7 apples, throws 1 apple, gives 4 apples to Bart and Bart gives one back:

7 -1 = 6

6 -4 = 2

2 +1 = 3

Alice has 5 apples, throws 3 apples, gives 2 to Bob and Bob gives one back, how many apples does Alice have?"

**Answer:** 1

RETURN TO SLIDE #17

# Demo #6 - C# talking to OpenAI with low level libraries.

dotnet new console -n OpenAiDirect  
cd OpenAiDirect  
dotnet add package OpenAI  
dotnet add package System.Configuration.ConfigurationManager

#### Add App.config.

<?xml version="1.0"?>

<configuration>

<appSettings>

<add key="key" value="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" />

</appSettings>

</configuration>

#### Add this cod to Program.cs:

using System.Configuration;

using OpenAI\_API;

using OpenAI\_API.Completions;

var key = ConfigurationManager.AppSettings["key"];

APIAuthentication aPIAuthentication = new APIAuthentication(key);

OpenAIAPI openAiApi = new OpenAIAPI(aPIAuthentication);

try {

string prompt = "Once upon a time";

Console.WriteLine("Starting prompt: " + prompt);

while (true) {

var completionRequest = new CompletionRequest {

Prompt = prompt,

Model = "gpt-3.5-turbo-instruct",

MaxTokens = 50

};

var completionResult = await openAiApi.Completions.CreateCompletionAsync(completionRequest);

var generatedText = completionResult.Completions[0].Text.Trim();

Console.WriteLine("Generated text:");

Console.WriteLine(generatedText);

Console.WriteLine("\n \*\*\*\* Press Enter to continue or type 'quit' to exit");

string input = Console.ReadLine()!;

if (input == "quit")

break;

else

prompt = prompt + " " + generatedText;

}

} catch (Exception ex) {

Console.WriteLine($"Error: {ex.Message}");

}

#### Run the app.

Type “quit” to stop.

# Demo #7 - C# talking to Azure OpenAI with low level libraries

Let us do the very same app with Azure OpenAI

dotnet new console -n OpenAiAzure  
cd OpenAiAzure  
dotnet add package Azure.AI.OpenAI --prerelease  
dotnet add package System.Configuration.ConfigurationManager

#### Add App.config.

<?xml version="1.0"?>

<configuration>

<appSettings>

<add key="endpoint" value="https://sweedencentralmze.openai.azure.com/" />

<add key="key" value="6c62b0b787fe495eb7de95716abc00a2" />

<add key="engine" value="gpt-35-turbo" />

</appSettings>

</configuration>

#### Add this cod to Program.cs:

using System.Configuration;

using Azure;

using Azure.AI.OpenAI;

string endpoint = ConfigurationManager.AppSettings["endpoint"]!;

string key = ConfigurationManager.AppSettings["key"]!;

string engine = ConfigurationManager.AppSettings["engine"]!;

var client = new OpenAIClient(new Uri(endpoint), new AzureKeyCredential(key));

try

{

string prompt = "Once upon a time";

Console.WriteLine("Starting prompt: " + prompt);

while (true)

{

CompletionsOptions completionsOptions = new()

{

DeploymentName = engine,

Prompts = { prompt },

};

Response<Completions> completionResult = client.GetCompletions(completionsOptions);

string generatedText = completionResult.Value.Choices[0].Text;

Console.WriteLine($"Chatbot: {generatedText}\n");

Console.WriteLine("\n \*\*\*\* Press Enter to continue or type 'quit' to exit");

string input = Console.ReadLine()!;

if (input == "quit")

break;

else

prompt = prompt + " " + generatedText;

}

}

catch (Exception ex)

{

Console.WriteLine($"Error: {ex.Message}");

}

#### Run the app.

The app behaves just the same as the one done using the API with OpenAI directly. Type “quit” to stop.

RETURN TO SLIDE #21

# Demo #8 - Build simple C# completion app with Azure OpenAI and Semantic Kernel Tool

<https://blog.medhat.ca/2024/01/build-simple-c-completion-app-with.html>

# Resources

Very important to watch this:

<https://www.youtube.com/playlist?list=PLmsFUfdnGr3zAgBMu4l1W713a0W__zAMl>

Support site:

<https://github.com/microsoft/generative-ai-for-beginners?WT.mc_id=academic-105485-koreyst>

**Good prompt engineering:**

<https://www.youtube.com/watch?v=R3sHRPP2G7A&list=PLmsFUfdnGr3zAgBMu4l1W713a0W__zAMl&index=3>

<https://www.youtube.com/watch?v=32GBH6BTWZQ&list=PLmsFUfdnGr3zAgBMu4l1W713a0W__zAMl&index=4>

First is better than second

**Basics about building apps**

<https://www.youtube.com/watch?v=5jKHzY6-4s8&list=PLmsFUfdnGr3zAgBMu4l1W713a0W__zAMl&index=5>