

Exploring the Power of Generative AI and Large Language Models (LLMs)

Richard Crane, Founder/CxO



M I L L 5



wickedprogrammer.com



richardcrane



richcrane

- › Founder/CTO of  **MILL5**
- › Worked at  Microsoft® for 10+ years
- › SQL Server Certified Master
- › Graduated Summa Cum Laude from  **Drexel** UNIVERSITY
- › Former Adjunct Professor at  **Drexel** UNIVERSITY
- › Authored *Essential Windows Communication Foundation* for Addison Wesley 
- › Wrote *Developing High-Quality Code, The .NET Way* for Software Development Times (<https://bit.ly/qualitycode>)

What is Artificial Intelligence?

- **Artificial Intelligence (AI)** is the **simulation of human intelligence** processes by machines, especially computer systems. It involves learning (acquiring and using information), reasoning (using rules to reach conclusions), and self-correction.
- AI technologies include machine learning, deep learning, natural language processing, and expert systems, with applications in various fields like autonomous vehicles, healthcare, gaming, and more.

What is Generative AI?

- **Generative AI** is a branch of artificial intelligence that uses machine learning to create new data similar to its training data. This new data can take the form of text, images, audio, or video. It's commonly used in fields like art, music, healthcare, and gaming.
- Example of Generative AI include ChatGPT, DALL-E 2, Deep Dream, GitHub CoPilot.

What is Machine Learning?

- **Machine Learning (ML)** is a branch of artificial intelligence (AI) that allows computers to learn to do things by themselves without giving them explicit instructions. Instead, they rely on patterns and inferences instead.
- Types of Machine Learning:
 - Supervised Learning: Uses labeled data to predict outcomes.
 - Unsupervised Learning: Finds structure or patterns in input data without labels.
 - Reinforcement Learning: An agent learns to behave by performing actions and observing the results. *
- **Deep Learning** is a type of machine learning that is even more advanced, using a group of interconnected brain-like cells known as neural networks to learn and understand complex things.

What is Natural Language Processing?

- **Natural Language Processing (NLP)** is a subfield of artificial intelligence that focuses on how computers can understand and interact with human language.
- NLP tasks include translating languages, recognizing speech, understanding sentiments, generating human-like text, and extracting information from text.
- It's used in various applications like voice assistants, translation services, and chatbots. Real world examples include Apple's Siri, Amazon Alexa, and Google Assistant.

What are Large Language Models?

- **Large Language Models (LLMs)** play a key role in NLP as they are used to generate human-like text. They are trained on vast amounts of text data, allowing them to learn patterns in a language, such as grammar, style, and context.
- LLMs models can perform a wide variety of NLP tasks such as machine translation, text summarization, sentiment analysis, and question answering, among others.
- GPT-3, GPT-4, and ChatGPT by OpenAI are examples of Large Language Models (LLMs).

Understanding ChatGPT

- One of the most popular LLMs in the world, but why?
- Training the model comes in a pretraining and fine-tuning step.
 - ... uses Self-Supervised learning during pretraining
 - ... uses Reinforcement Learning from Human Feedback
 - ... along with reward modeling
- Leverages Google's Transformer architecture which is a type of neural network used for creating the model using Deep Learning.
 - "Attention is All You Need," published by Vaswani et al. in 2017
- Uses a very large amount of data and parameters to train the models, especially with GPT-4.

Models vs. Parameters

Model	Parameters	Owner
GPT-3	175 Billion	OpenAI
BigScience Large Open-science Open-access Multilingual Language Model (BLOOM)	176 Billion	Hugging Face, BigScience
ESMFold	15 Billion	Meta AI
Gato	79M, 364M, and 1.18 Billion	DeepMind
Language Model for Dialogue Application (LaMDA)	137 Billion	Google
Megatron-Turing Natural Language Generation (MT-NLG)	530 Billion	Microsoft, NVidia
Wu Dao 2.0	1.75 Trillion	Beijing Academy of Artificial Intelligence
GPT-4	170 Trillion (Rumored)	Open AI

<https://aibusiness.com/nlp/7-language-models-you-need-to-know>

Deep Dream and DALL-E

DEMO



Deep Dream vs. DALL-E

Generate a realistic photo that is a combination of a Marvel SuperHero and Einstein.

Some Real-World Use Cases of AI (MILL5)

- Vision Detection in the Surgical Room
 - Detect and analyze the state of the operating room.
- Persona Analysis and Optimization for Trading Systems
 - Goal is to understand different types of personas of traders and tailor the experience to the user.
- Sentiment Analysis
 - Determine the sentiment of text and speech for help in making better business decisions.
- Gaming and Entertainment
 - Generate content tailored to the user for gamification.

Analyzing Operating Room in Real Time

<https://www.youtube.com/watch?v=3RENSMNdKDY>

VIDEO



How do I integrate ChatGPT4 into my application



Integrating ChatGPT-4 into your application would typically involve using the OpenAI API to call the model and receive responses. As of my knowledge cutoff in September 2021, OpenAI's GPT-3 was the latest model accessible through an API, and instructions for accessing and using it were available in the OpenAI documentation.



Although specific instructions may have changed since then, here's the general process for integrating a GPT model into an application based on the last accessible information:

Integrating ChatGPT4 into Your Application
Let's Ask ChatGPT

DEMO

an account with OpenAI, which will

your API key.

Integrating ChatGPT4 into Your Application (Python)



To integrate ChatGPT-4 into your application:

1. **Sign up for OpenAI and get your API key.**
2. **Install the necessary packages.** If using Python, run `pip install openai`.
3. **Use the API key to make requests to OpenAI's API.** In Python:

python

Copy code

```
import openai

openai.api_key = 'your-api-key'

response = openai.ChatCompletion.create(
    model="gpt-4.0-turbo",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Who won the world series in 2020?"},
    ]
)

print(response['choices'][0]['message']['content'])
```

Replace 'your-api-key' with your actual key, and 'gpt-4.0-turbo' with the correct model identifier.

4. **Parse the response** to display it in your application.

Remember to check the latest OpenAI documentation for up-to-date details. Also, understand the cost, rate limits, and use case policy of OpenAI.

Integrating ChatGPT4 into Your Application (C#)

Here's an example of how you can use `System.Text.Json` to integrate ChatGPT-4 in your C# application:

csharp

Copy code

```
using System;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Text.Json;

public class Program
{
    public static async Task Main()
    {
        var client = new HttpClient();
        var openaiUrl = "https://api.openai.com/v1/chat/completions";
        var apiKey = "your-api-key";

        var payload = new
        {
            model = "gpt-4.0-turbo",
            messages = new object[]
            {
                new { role = "system", content = "You are a helpful assistant" },
                new { role = "user", content = "Who won the world series in 2023?" }
            }
        };

        client.DefaultRequestHeaders.Add("Authorization", $"Bearer {apiKey}");
        var response = await client.PostAsync(openaiUrl,
            new StringContent(JsonSerializer.Serialize(payload), Encoding.UTF8));

        var responseObject = JsonSerializer.Deserialize<dynamic>(await response.Content.ReadAsStringAsync());
        Console.WriteLine(responseObject["choices"][0]["message"]["content"]);
    }
}
```


Introducing Azure OpenAI Studio and Service

- Must have an Azure subscription and apply to enable access
 - Apply now so that you can get through the waiting list
 - <https://aka.ms/oaiapply>
- Azure OpenAI Studio
 - Development platform for developing AI solutions
 - Model catalog for leveraging models such chatgpt35-turbo, chatgpt4
 - Access to training capabilities for
- Azure OpenAI Service
 - Set of REST services integrating your application

Azure AI

Applications



Microsoft 365



Microsoft
Dynamics 365

Partner Solutions

Application Platform

AI Builder



Power BI



Power Apps



Power Automate



Power Virtual Agents

Scenario-Based Services



Bot Service



Cognitive Search



Form Recognizer



Video Indexer



Metrics Advisor



Immersive Reader

Customizable AI Models



Vision



Speech



Language



Decision



Azure OpenAI Service

ML Platform



Azure Machine Learning

Radically changing the art of possible with Azure OpenAI Service

Large pretrained foundation AI models
custom-tunable with your parameters and your data



Summarization
Reasoning over data



Writing tools
Code generation



ChatGPT
The Era of Copilots



GPT-3 (GA)

DALL•E 2 (preview)

ChatGPT (GA)

GPT-4 (GA)

Foundation of enterprise security, privacy and compliance

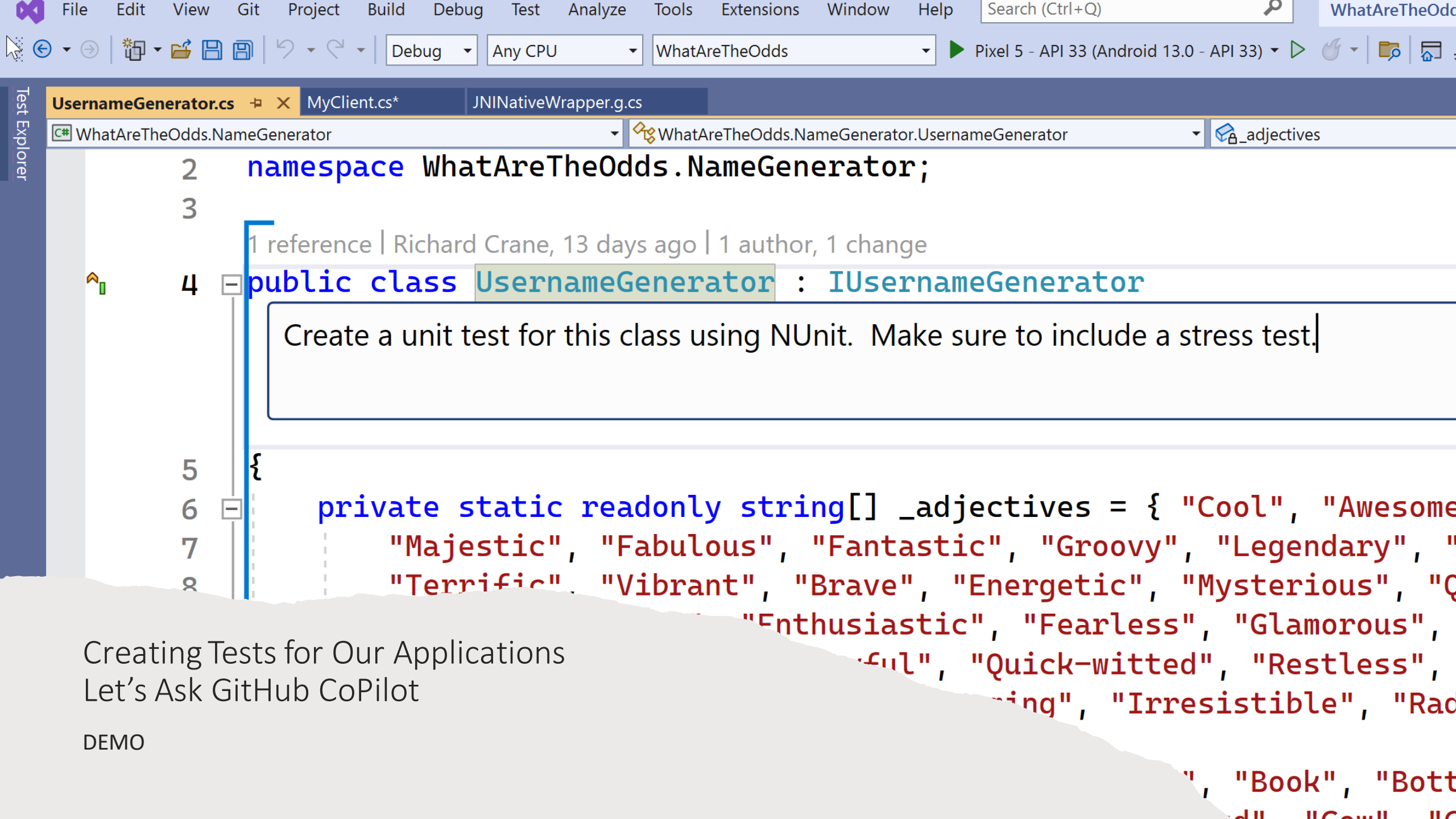
Model Selection

- Picking your model becomes a matter of cost and capability.
- Select the models you need.
 - <https://platform.openai.com/docs/models/>
 - <https://learn.microsoft.com/en-us/azure/cognitive-services/openai/concepts/models>
- You may not need to the most expensive model.
 - Ex. Can you use chatgpt-35-turbo vs. gpt4?
- Recommendation is to start with the most capable model to meet your needs and then move downward to reduce costs. Each time, you reevaluate whether the new model still meets your needs.

Real World Integration of AI

- Expose microservices that leverage OpenAI or Azure OpenAI.
- Remember that using OpenAI costs money (\$\$\$).
 - Cost based on tokens which add up
 - Rate limit your APIs
 - Cache your APIs *
- Remember that people do evil things.
 - Secure your APIs
 - Rate limit your APIs
 - Protect your APIs with ban lists
 - Protect your APIs with allow lists

* Remember that with AI you don't get a different response each time.



UsernameGenerator.cs

MyClient.cs*

JNINativeWrapper.g.cs

WhatAreTheOdds.NameGenerator

WhatAreTheOdds.NameGenerator.UsernameGenerator

_adjectives

```
2 namespace WhatAreTheOdds.NameGenerator;
```

```
3
```

1 reference | Richard Crane, 13 days ago | 1 author, 1 change

```
4 public class UsernameGenerator : IUsernameGenerator
```

Create a unit test for this class using NUnit. Make sure to include a stress test.

```
5 {
```

```
6     private static readonly string[] _adjectives = { "Cool", "Awesome"
```

```
7         "Majestic", "Fabulous", "Fantastic", "Groovy", "Legendary", "
```

```
8         "Terrific", "Vibrant", "Brave", "Energetic", "Mysterious", "Q
```

```
        "Enthusiastic", "Fearless", "Glamorous",
```

```
        "Successful", "Quick-witted", "Restless",
```

```
        "Amazing", "Irresistible", "Rad
```

```
        "Book", "Bott
```

```
        "d", "Cow", "
```

Creating Tests for Our Applications
Let's Ask GitHub CoPilot

DEMO

Unit tests for including stress test using GitHub CoPilot



Richard Crane

Create a unit test for this class using NUnit. Make sure to include a stress test.



Copilot

```
using NUnit.Framework;
using System.Collections.Generic;

public class UsernameGeneratorTests
{
    private UsernameGenerator _usernameGenerator;

    [SetUp]
    public void Setup()
    {
        _usernameGenerator = new UsernameGenerator();
    }

    [Test]
    public void GenerateUsername_ReturnsAString()
    {
        string generatedUsername = _usernameGenerator.GenerateUsername();

        Assert.IsInstanceOf<string>(generatedUsername);
    }

    [Test]
    public void GenerateUsername_ReturnsUniqueUsernames()
    {
        List<string> generatedUsernames = new List<string>();

        for (int i = 0; i < 1000; i++)
        {
            string generatedUsername = _usernameGenerator.GenerateUsername();

            Assert.IsFalse(generatedUsernames.Contains(generatedUsername));
            generatedUsernames.Add(generatedUsername);
        }
    }
}
```

Ask Copilot



Quick Take Away

- *“Artificial Intelligence (AI) and the use of Large Language Models such as ChatGPT is helping to accelerate our learning each day. If you are not using AI, you need to start doing so immediately. It will be the difference on whether you take yourself to the next level or get left behind.” – Richard Crane*
- *“The application of Artificial Intelligence (AI) and the utilization of advanced language models like ChatGPT are increasingly enhancing our daily learning experiences. If you have yet to embrace AI, now is the time to do so without delay. It can make all the difference in propelling you to new heights or risk falling behind in today's rapidly evolving landscape.” – ChatGPT35*
- *“Utilizing Artificial Intelligence (AI), including Large Language Models like ChatGPT, is enhancing our learning processes every day. Those who have yet to embrace AI should consider incorporating it into their routine without delay. It could prove to be the determining factor between advancing to new heights and lagging behind.” – ChatGPT4*
- *“AI and Large Language Models like ChatGPT boost daily learning. Not using AI? Start now to avoid being left behind and ensure progress.” – ChatGPT4 (Shorter)*

Summary and Q&A

- This presentation was created using ChatGPT4.

Thank You!



For more information about MILL5, visit www.mill5.com.

Connect with us on social!

