Microsoft

Microsoft Azure Virtual Training Day: Generative AI Fundamentals

Explore fundamentals of Generative AI

Microsoft

Learning Objectives

- **Explore fundamentals of Generative AI**
  - What is generative AI?
  - Large language models
  - What is Azure OpenAI?
  - What are copilots?
  - Improve generative AI responses with prompt engineering

Microsoft

Learning Objective: Explore fundamentals of Generative AI

What is generative AI?

Artificial Intelligence (AI) imitates human behavior by using machine learning to interact with the environment and execute tasks without explicit directions on what to output.

Generative AI describes a category of capabilities within AI that create original content. People typically interact with generative AI that has been built into chat applications.

Generative AI applications take in natural language input, and return appropriate responses in a variety of formats:

- Natural language generation
- Image generation
- Code generation

Large language models

Generative AI applications are powered by *large language models* (LLMs), which are a specialized type of machine learning model that you can use to perform natural language processing (NLP) tasks, including:

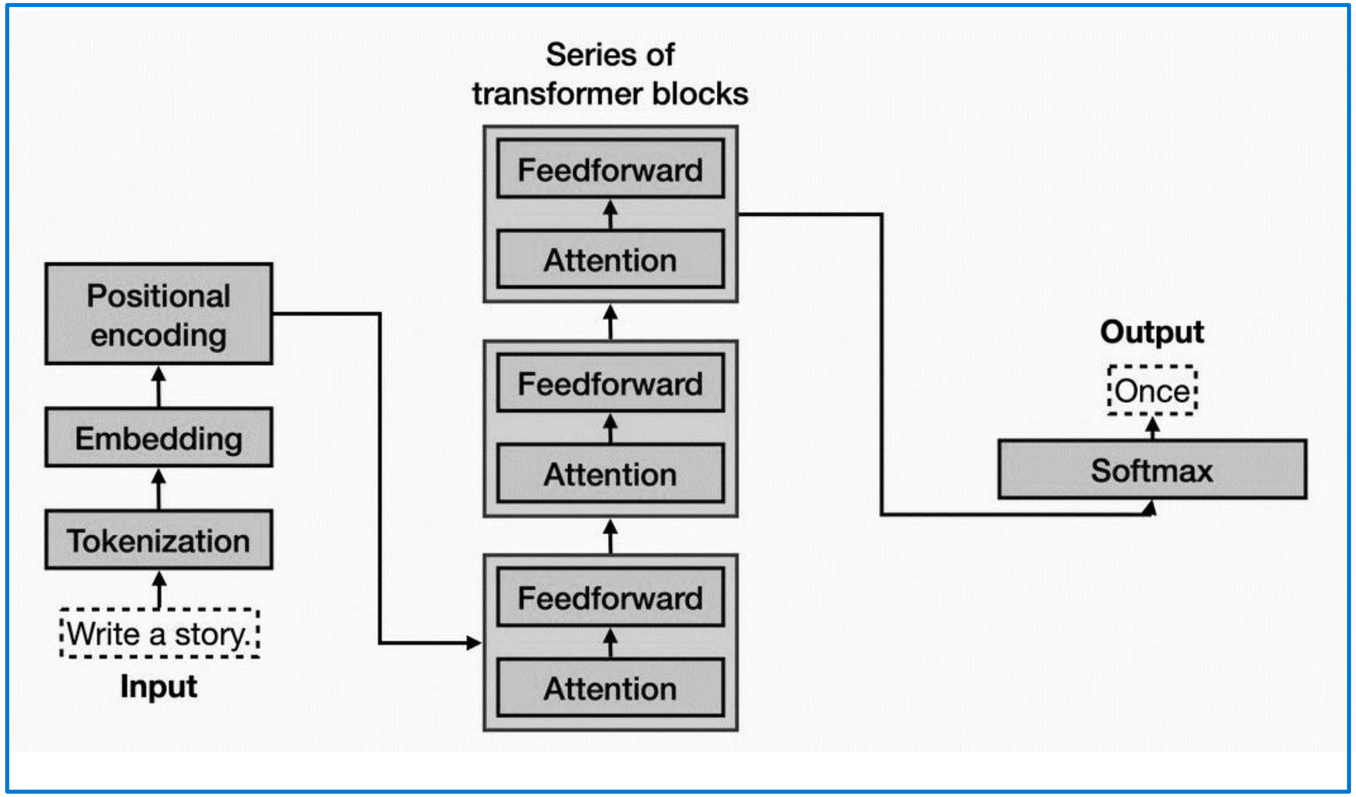
- Determining *sentiment* or otherwise classifying natural language text.
- Summarizing text.
- Comparing multiple text sources for semantic similarity.
- Generating new natural language.

Large language models

Transformer models

Transformer model architecture consists of two components, or *blocks*:

- An ***encoder*** block that creates semantic representations of the training vocabulary
- A ***decoder*** block that generates new language sequences



Large language models – Tokenization

Step one: tokenization

The first step in training a transformer model is to decompose the training text into *tokens*.

Example sentence: *I heard a dog bark loudly at a cat.*

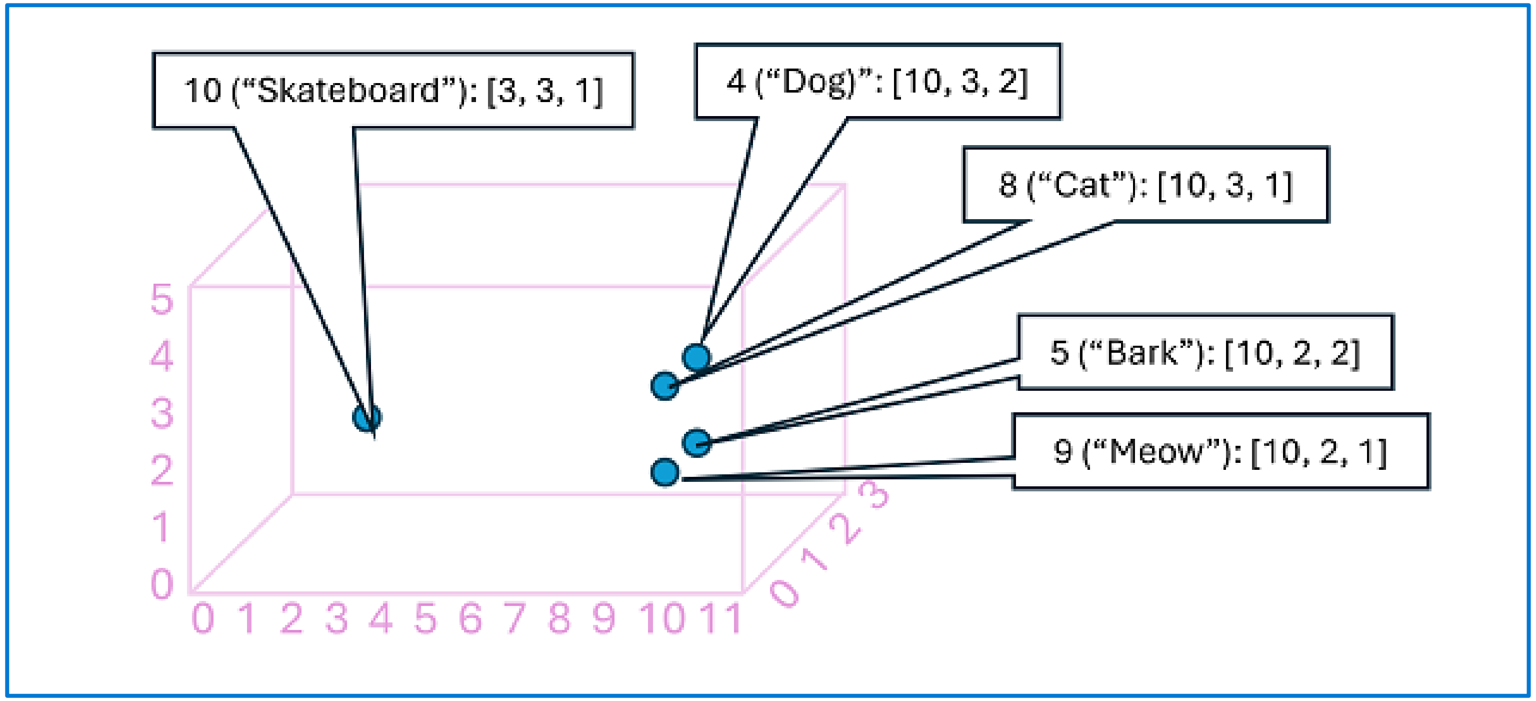
"I"=1	"heard"=2	"a"=3	"dog"=4	"bark"=5	"loudly"=6	"at"=7	"cat"=8

- The sentence is now represented with the tokens: *[1 2 3 4 5 6 7 3 8]*.
- Note "a" is tokenized as 3 only once.
- Similarly, the sentence "I heard a cat" could be represented with the tokens *[1 2 3 8]*.

Large language models – Embeddings

Step two: embeddings

The relationships between tokens are captured as vectors, known as embeddings.



Token	Word	Embedding
10	Skateboard	[3, 3, 1]
4	Dog	[10,3,2]
8	Cat	[10,3,1]
5	Bark	[10,2,2]
9	Meow	[10,2,1]

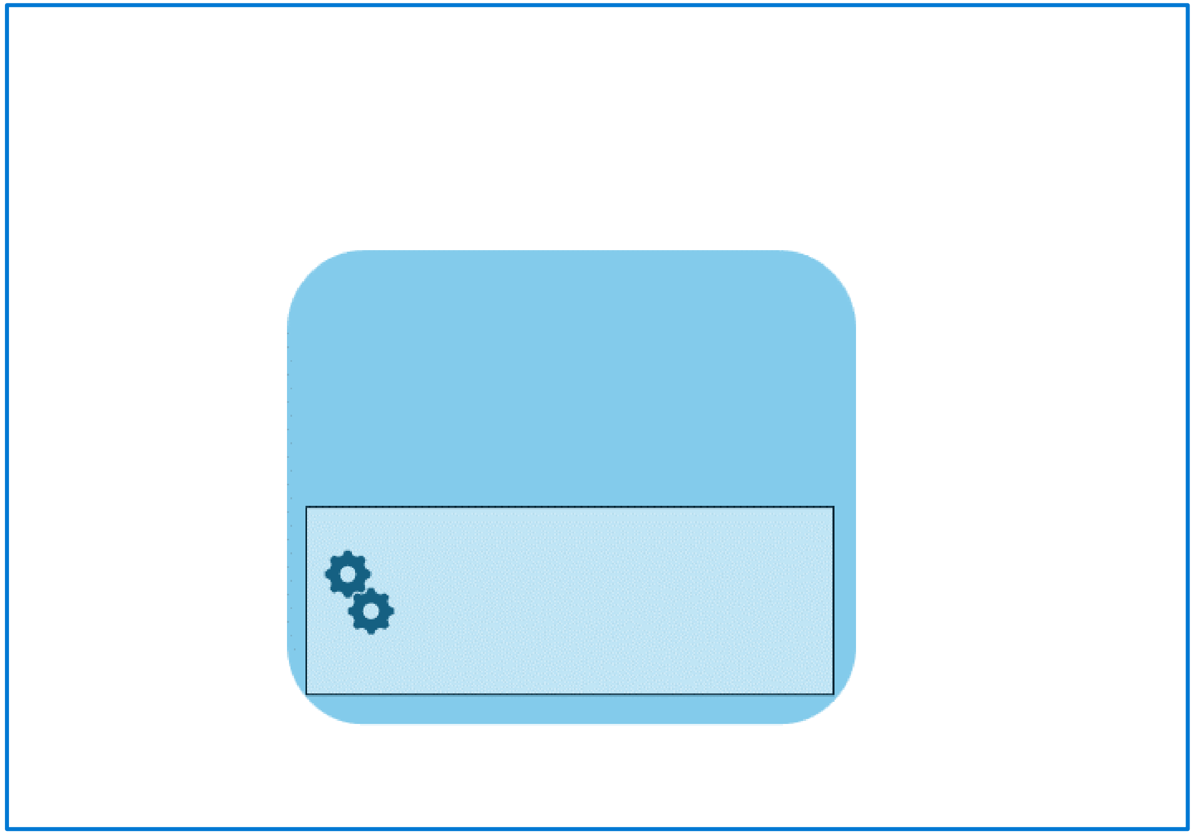
Large language models – Attention

Step three: attention

Capture the strength of relationships between tokens using the attention technique.

Example:

- Goal: predict token after “**dog**”
- Represent “**I heard a dog**” as vectors
- Assign “**heard**” and “**dog**” more weight
- Several possible tokens can come after dog
- The most probable token is added to the sequence, in this case “**bark**”.



What is Azure OpenAI?

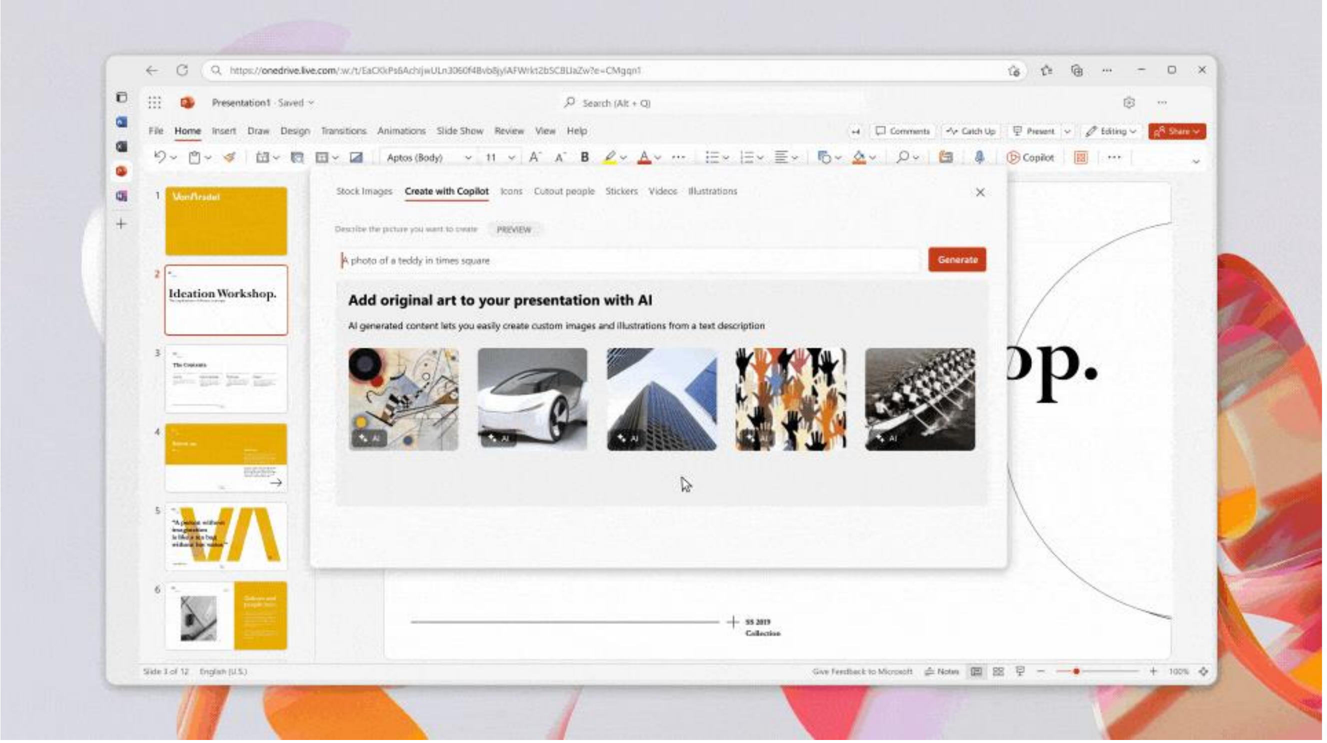
Azure OpenAI Service is Microsoft's cloud solution for deploying, customizing, and hosting large language models.

Azure OpenAI supports many LLMs:	**Description**
GPT-4	A set of models that improve on GPT-3.5 and can understand as well as generate natural language and code.
GPT-3.5	A set of models that improve on GPT-3 and can understand as well as generate natural language and code.
Embeddings	A set of models that can convert text into numerical vector form to facilitate text similarity.
DALL-E (Preview)	A series of models in preview that can generate original images from natural language.

What are copilots?

Copilots are often integrated into other applications and provide a way for users to get help with common tasks from a generative AI model.

- Developers can build copilots that submit prompts to large language models and generate content for use in applications.
- Business users can use copilots to boost their productivity and creativity with AI-generated content.



A photo of a teddy in times square

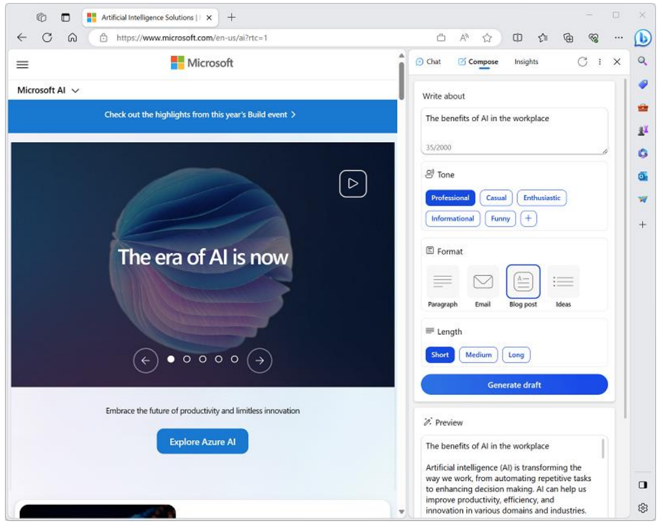
Add original art to your presentation with AI

AI generated content lets you easily create custom images and illustrations from a text description

What are copilots?

Other examples of copilots

Microsoft Edge browser copilot



Check out the highlights from this year's Build event >

Embrace the future of productivity and limitless innovation

Write about: The benefits of AI in the workplace

35/2000

Tone: Professional (selected), Casual, Enthusiastic, Informational, Funny, +

Format: Blog post

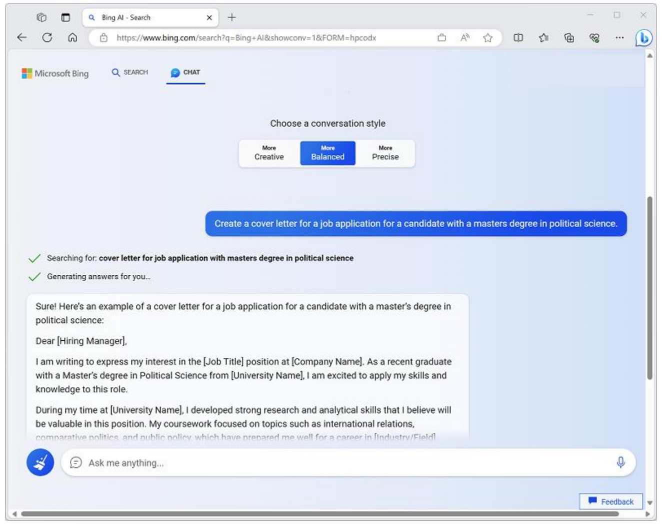
Length: Short (selected), Medium, Long

Preview

The benefits of AI in the workplace

Artificial intelligence (AI) is transforming the way we work, from automating repetitive tasks to enhancing decision making. AI can help us improve productivity, efficiency, and innovation in various domains and industries.

Microsoft Bing copilot



Choose a conversation style

More Creative | More Balanced (selected) | More Precise

Create a cover letter for a job application for a candidate with a masters degree in political science.

✓ Searching for: **cover letter for job application with masters degree in political science**

✓ Generating answers for you...

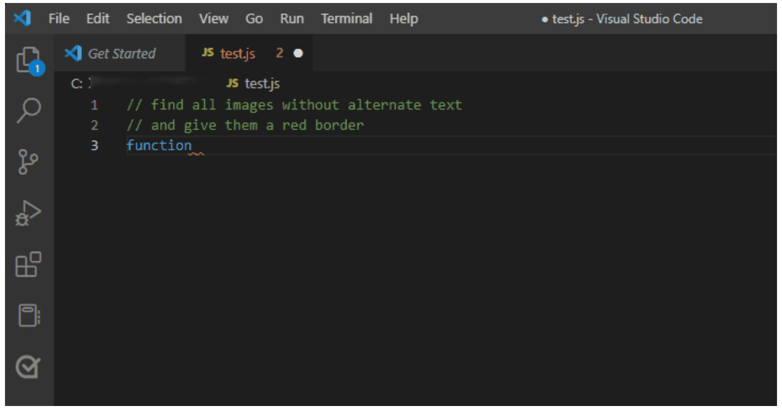
Sure! Here's an example of a cover letter for a job application for a candidate with a master's degree in political science:

Dear [Hiring Manager],

I am writing to express my interest in the [Job Title] position at [Company Name]. As a recent graduate with a Master's degree in Political Science from [University Name], I am excited to apply my skills and knowledge to this role.

During my time at [University Name], I developed strong research and analytical skills that I believe will be valuable in this position. My coursework focused on topics such as international relations, comparative politics, and public policy, which have prepared me well for a career in [Industry/Field]

GitHub copilot



```
// find all images without alternate text
// and give them a red border
function
```

And many more!

Improve generative AI responses with prompt engineering

The term *prompt engineering* describes the process of prompt improvement.

Both developers who design applications and consumers who use applications can improve the quality of responses from generative AI by using direct language, system messages, examples, and/or grounding data.

	Description	Example
Direct language	You can get the most useful completions by being explicit about the kind of response you want	"**Create a list** of 10 things to do in Edinburgh during August"
System messages	Describe how the chat should act	"You're a **helpful** assistant that **responds in a cheerful, friendly** manner"
Providing examples	LLMs generally support *zero-shot learning* in which responses can be generated without prior examples. However, you can also provide a few example responses, known as *few-shot learning*	"Visit the castle in the morning before the crowds arrive"
Grounding data	You can include *grounding* data to provide context	Including **email text** with the prompt "Summarize my email"

Demo

- Explore generative AI with Bing Chat copilot

Introduction to Azure OpenAI Service

Microsoft

Learning Objectives

- **Introduction to Azure OpenAI Service**
  - What is generative AI
  - Describe Azure OpenAI
  - How to use Azure OpenAI
  - Understand OpenAI's natural language capabilities
  - Understand OpenAI code generation capabilities
  - Understand OpenAI's image generation capabilities
  - Describe Azure OpenAI's access and responsible AI policies

Microsoft

Learning Objective:
Introduction to Azure OpenAI Service

What is generative AI?

Artificial Intelligence (AI) imitates human behavior by using machine learning to interact with the environment and execute tasks without explicit directions on what to output.

Generative AI describes a category of capabilities within AI that create original content. People typically interact with generative AI that has been built into chat applications.

Generative AI applications take in natural language input, and return appropriate responses in a variety of formats:

- Natural language generation
- Image generation
- Code generation

What is Azure OpenAI?

Azure OpenAI service is Microsoft's cloud solution for deploying, customizing, and hosting large language models.

Azure OpenAI service consists of:

- Pre-trained generative AI models
- Customization capabilities
- Built-in tools to detect and mitigate harmful use cases so users can implement AI responsibly
- Enterprise-grade security with role-based access control (RBAC) and private networks

You can use several methods to develop Azure OpenAI solutions: Azure AI Studio, REST API, supported SDKs, and Azure CLI.

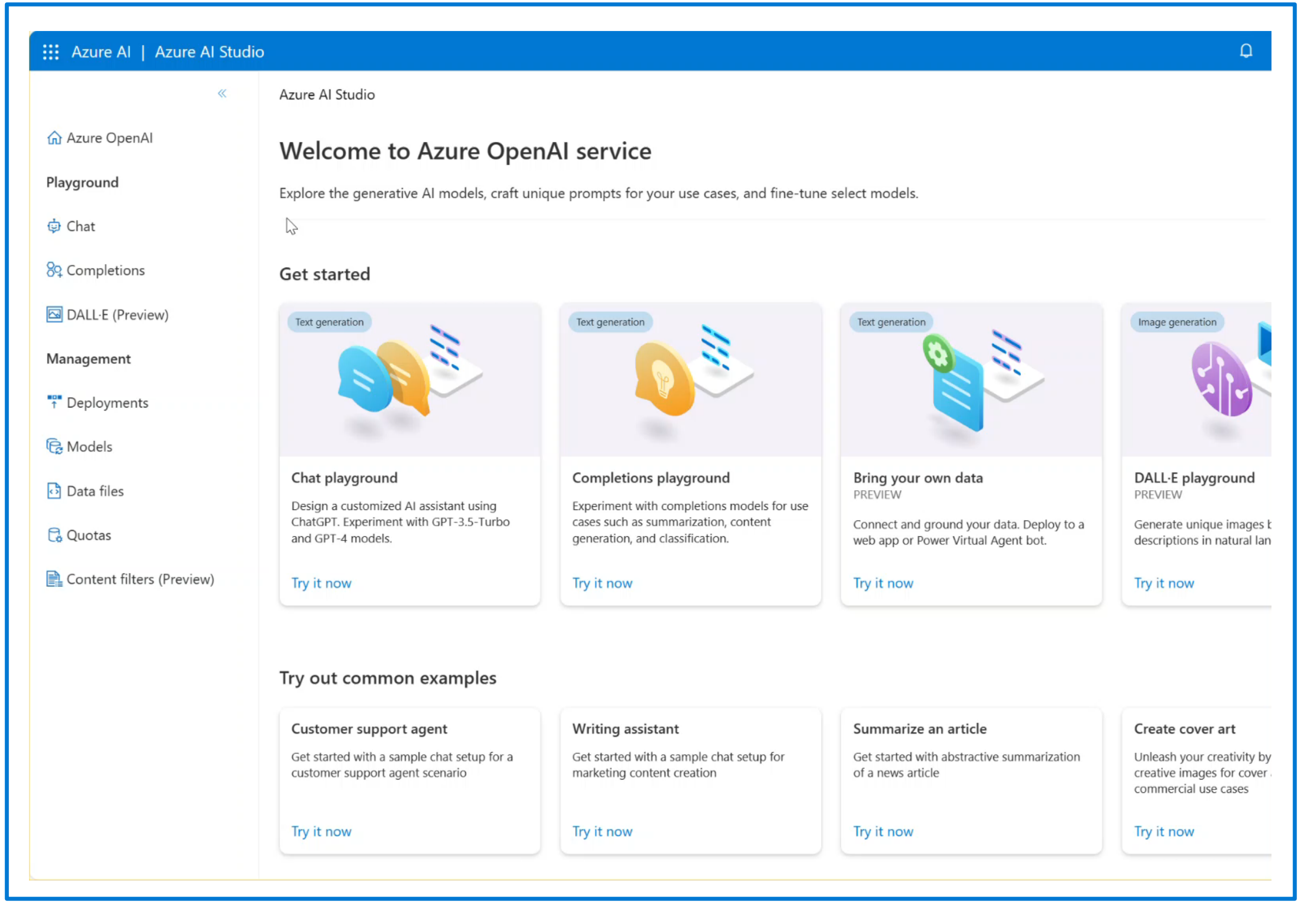
How to use Azure OpenAI

Azure OpenAI Studio:

- Build and deploy AI models for software applications
- Powered by generative AI models optimized for various tasks
- Azure OpenAI models include: **GPT-4**, **GPT-3.5**, **Embeddings** and **DALL-E models**

Playgrounds:

- Experiment with OpenAI models without coding
- Use the assistant setup to instruct the model about how it should behave



Welcome to Azure OpenAI service

Explore the generative AI models, craft unique prompts for your use cases, and fine-tune select models.

Get started

Chat playground

Design a customized AI assistant using ChatGPT. Experiment with GPT-3.5-Turbo and GPT-4 models.

Completions playground

Experiment with completions models for use cases such as summarization, content generation, and classification.

Bring your own data
PREVIEW

Connect and ground your data. Deploy to a web app or Power Virtual Agent bot.

DALL·E playground
PREVIEW

Generate unique images b
descriptions in natural lan

Try out common examples

Customer support agent

Get started with a sample chat setup for a customer support agent scenario

Writing assistant

Get started with a sample chat setup for marketing content creation

Summarize an article

Get started with abstractive summarization of a news article

Create cover art

Unleash your creativity by
creative images for cover
commercial use cases

Understand OpenAI's natural language capabilities

Generative pre-trained transformer (GPT) models are excellent at both understanding and creating natural language.

For example, given a prompt where the user types in text asking for a cooking recipe:

```
Write recipe instructions for a fruit pie
based on these ingredients:

Strawberries
Blueberries
Flour
Eggs
Milk
```

Here's an example of the OpenAI API generated response:

```
Instructions:
1. Preheat your oven to 350 degrees Fahrenheit.
2. In a medium bowl, mix together 1 cup of flour and 1
egg until it forms a dough.
3. Grease a 9-inch pie dish with butter or non-stick
cooking spray.
4. Roll out the dough and place it in the pie dish.
5. In a separate bowl, mix together 1 cup of strawberries,
1 cup of blueberries, and 1 cup of milk.
6. Pour the fruit mixture into the pie dish.
7. Bake
```

Understand OpenAI code generation capabilities

GPT models translate natural language or code snippets into code.

Code generation goes beyond just writing code from natural language prompts. Given the following code, it can generate unit tests as shown on the right:

Python

```python
# Python 3
def mult_numbers(a, b):
  return a * b

# Unit test
def
```

Python

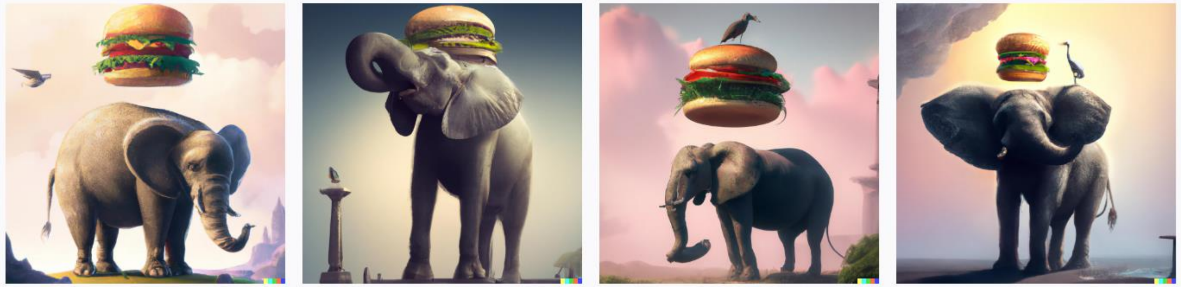
```python
# Python 3
def mult_numbers(a, b):
  return a * b

# Unit test
def test_mult_numbers():
  assert mult_numbers(3, 4) == 12
  assert mult_numbers(0, 10) == 0
  assert mult_numbers(4, 0) == 0

# Unit test
def test_mult_numbers_negative():
  assert mult_numbers(-1, 10) == -10
  assert mult_numbers(10, -1) == -10
```

Understand OpenAI's image generation capabilities

Generative AI models can edit and create images. The model that works with images is called DALL-E, which supports image creation, image editing and image variations creation.

- **Image generation:** With DALL-E, you can even request an image in a particular style. Styles can be used for edits and variations as well.
- **Editing an image:** DALL-E can edit the image as requested by changing its style, adding or removing items, or generating new content to add.
- **Image variations:** Image variations can be created by providing an image and specifying how many variations of the image you would like.

Prompt: Create four variations of an image of an elephant with a hamburger.



Demo

- Explore Azure OpenAI Service

Explore Responsible Generative AI

Microsoft

Learning Objectives

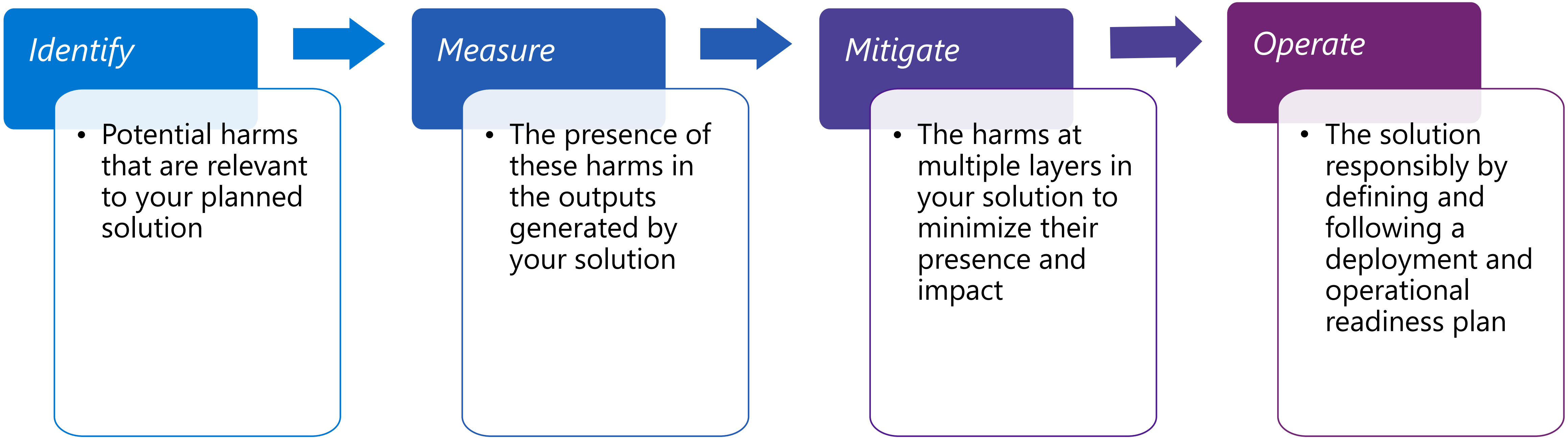
- **Explore Responsible Generative AI**
  - Plan a responsible generative AI solution:
    - Identify potential harms
    - Measure potential harms
    - Mitigate potential harms
    - Operate a responsible generative AI solution

Microsoft

Learning Objective: Explore Responsible Generative AI

Plan a responsible generative AI solution

Four stage process to develop and implement a plan for responsible AI are:



Identify

- Potential harms that are relevant to your planned solution

Measure

- The presence of these harms in the outputs generated by your solution

Mitigate

- The harms at multiple layers in your solution to minimize their presence and impact

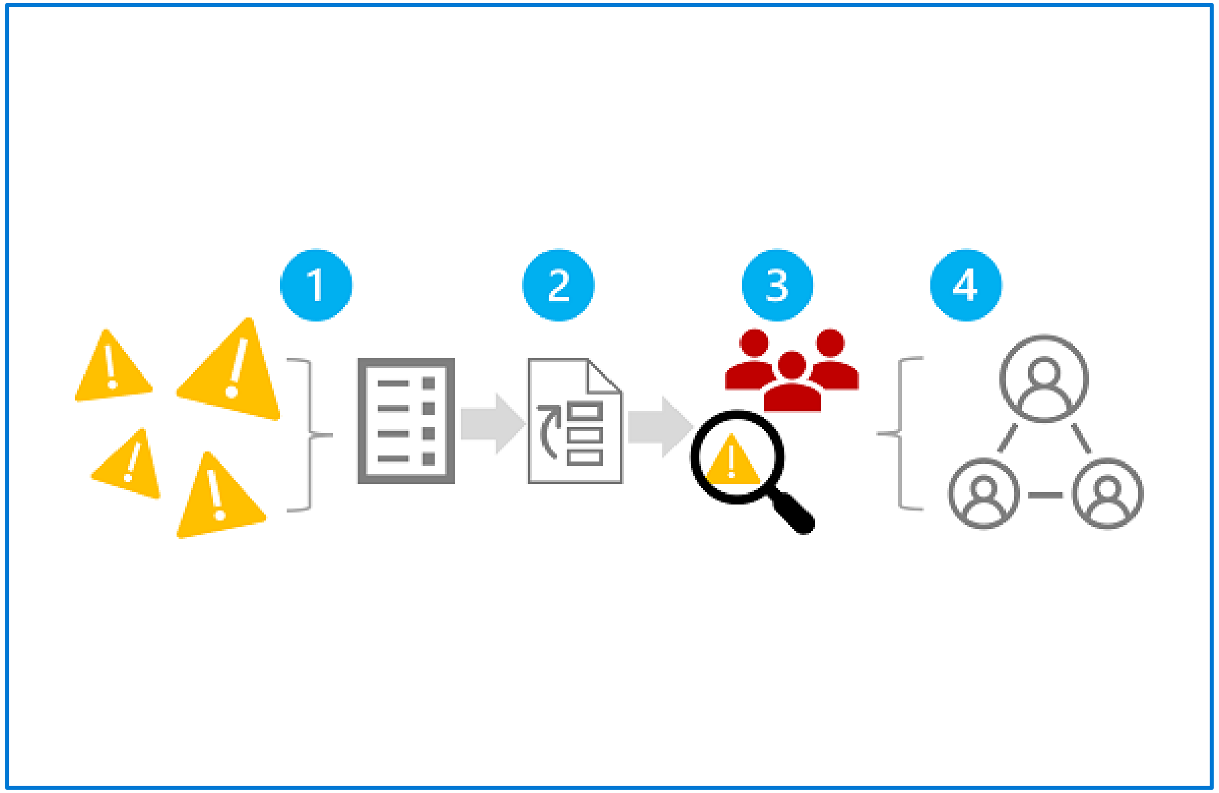
Operate

- The solution responsibly by defining and following a deployment and operational readiness plan

Identify potential harms

There are four steps in the 1st stage:

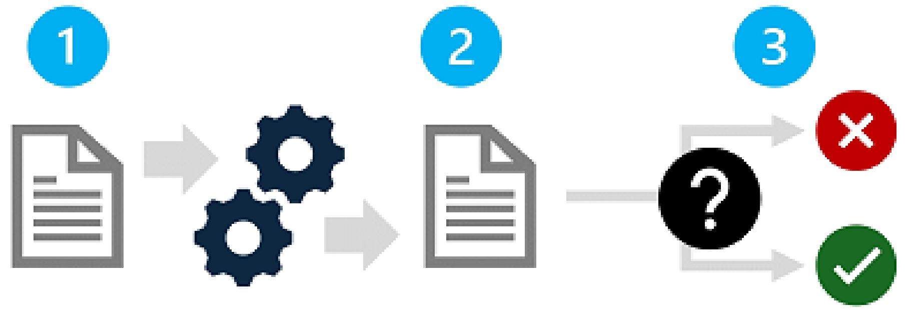
1. Identify potential harms
2. Prioritize identified harms
3. Test and verify the prioritized harms
4. Document and share the verified harms



Measure potential harms

Generalized approach for measuring potential harms involves three steps:

1. Prepare diverse input prompts for identified potential harms
2. Submit prompts, retrieve system-generated output
3. Apply pre-defined criteria to categorize output by harm level

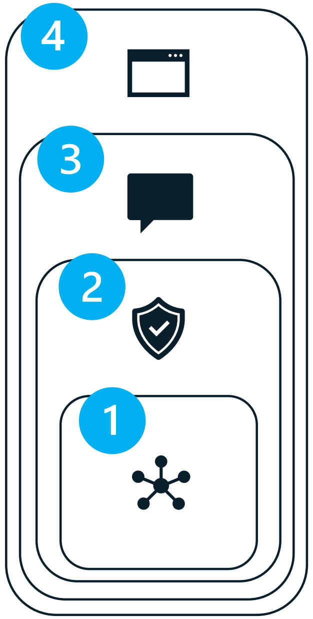


Classified as Microsoft Confidential

Mitigate potential harms

Mitigation of potential harms in a generative AI solution involves a layered approach, in which mitigation techniques can be applied at each of four layers.

Mitigation at the:



4. User experience level

3. Metaprompt and grounding level

2. Safety System level

1. Model level

Operate a responsible generative AI solution

A successful release requires some planning and preparation. Consider the following guidelines:

- *Phased delivery plan* for feedback: Release to restricted users first.
- *Incident response plan*: Estimating response time for unforeseen issues.
- *Rollback plan*: Steps to revert solution in case of incidents.
- Immediate harmful response blocking capability.
- Blocking users/applications/IPs for misuse prevention.
- User feedback/reporting mechanism for issues.
- Telemetry tracking for user satisfaction and privacy compliance.

Demo

- Explore content filters in Azure OpenAI Service