

Fundamentals of Generative AI

- 1 hr 13 min
- Module
- 11 Units

In this module, you explore the way in which language models enable AI applications and services to generate original content based on natural language input. You also learn how generative AI enables the creation of copilots that can assist humans in creative tasks.

Learning objectives

By the end of this module, you are able to:

- Understand generative AI's place in the development of artificial intelligence.
- Understand language models and their role in intelligent applications.
- Describe examples of copilots and good prompts.

Prerequisites

- Familiarity with Azure and the Azure portal.

This module is part of these learning paths

- [Copilot Foundations](#)
- [Get started with Microsoft Copilot for Security](#)
- [Microsoft Azure AI Fundamentals: Generative AI](#)
- [SC-200: Mitigate threats using Microsoft Copilot for Security](#)

Introduction

Completed 100 XP

- 1 minute

Generative AI, and technologies that implement it like Microsoft Copilot are increasingly in the public consciousness – even among people who don't work in technology roles or have a background in computer science or machine learning. The futurist and novelist Arthur C. Clarke is quoted as observing that "any sufficiently advanced technology is indistinguishable from magic." In the case of generative AI there does indeed seem to be an almost miraculous ability to produce human-like original content, including poetry, prose, and even computer code.

However, there's no wizardry involved in generative AI – just the application of mathematical techniques incrementally discovered and refined over many years of research into statistics, data science, and machine learning. You can gain a high-level understanding of how the magic trick is done by learning the core concepts and principles explored in this module. As you learn more about the generative AI technologies we have today, you can help society imagine new possibilities for AI tomorrow.

What is generative AI?

Completed 100 XP

- 2 minutes

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. One popular example of such an application is [Microsoft Copilot](#), a chatbot companion to browse the web more effectively.

Generative AI applications take in natural language input, and return appropriate responses in a variety of formats such as natural language, images, or code.

Natural language generation

To generate a natural language response, you might submit a request such as "*Write a cover letter for a person with a bachelor's degree in history.*"

A generative AI application might respond to such a request like this:

Dear Hiring Manager, I am writing to express my interest in the position of...

The screenshot shows a web browser window for https://copilot.microsoft.com. At the top, there is a Microsoft logo and a search bar containing the URL. Below the search bar, a blue button reads "Write a cover letter for a person with a bachelors degree in history". A list of two items follows, each preceded by a green checkmark: "Searching for: **cover letter for bachelors degree in history**" and "Generating answers for you...". A large text box contains a sample cover letter. The letter begins with "Dear Hiring Manager," followed by a paragraph about the applicant's recent graduation from a Bachelor's degree in History and their confidence in their academic background and skills. Another paragraph discusses their time at [university name] and how it contributed to their deep understanding of historical events and research skills. At the bottom of the text box are icons for thumbs up, thumbs down, a reply, and download. In the bottom right corner of the page, it says "1 of 30" with a small green dot.

Write a cover letter for a person with a bachelors degree in history

✓ Searching for: **cover letter for bachelors degree in history**

✓ Generating answers for you...

Sure, I can help you with that. Here's a sample cover letter for a person with a Bachelor's degree in History:

Dear Hiring Manager,

I am writing to express my interest in the position of [position name] at [company name]. As a recent graduate with a Bachelor's degree in History, I am confident that my academic background and skills make me a strong candidate for this role.

During my time at [university name], I developed a deep understanding of historical events and their impact on society. I honed my research and analytical skills through various coursework and projects, which I believe will be valuable in this position. Additionally, I have experience working in

1 of 30

Image generation

Some generative AI applications can interpret a natural language request and generate an appropriate image. For example, you might submit a request like "*Create a logo for a florist business.*"

A generative AI application could then return an original new image based on the description you provided, like this:



Code generation

Some generative AI applications are designed to help software developers write code. For example, you could submit a request like "*Write Python code to add two numbers.*" and generate the following response:

PythonCopy

```
def add_numbers(a, b):  
    return a + b
```

What are language models?

Completed 100 XP

- 8 minutes

Generative AI applications are powered by *language models*, 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.

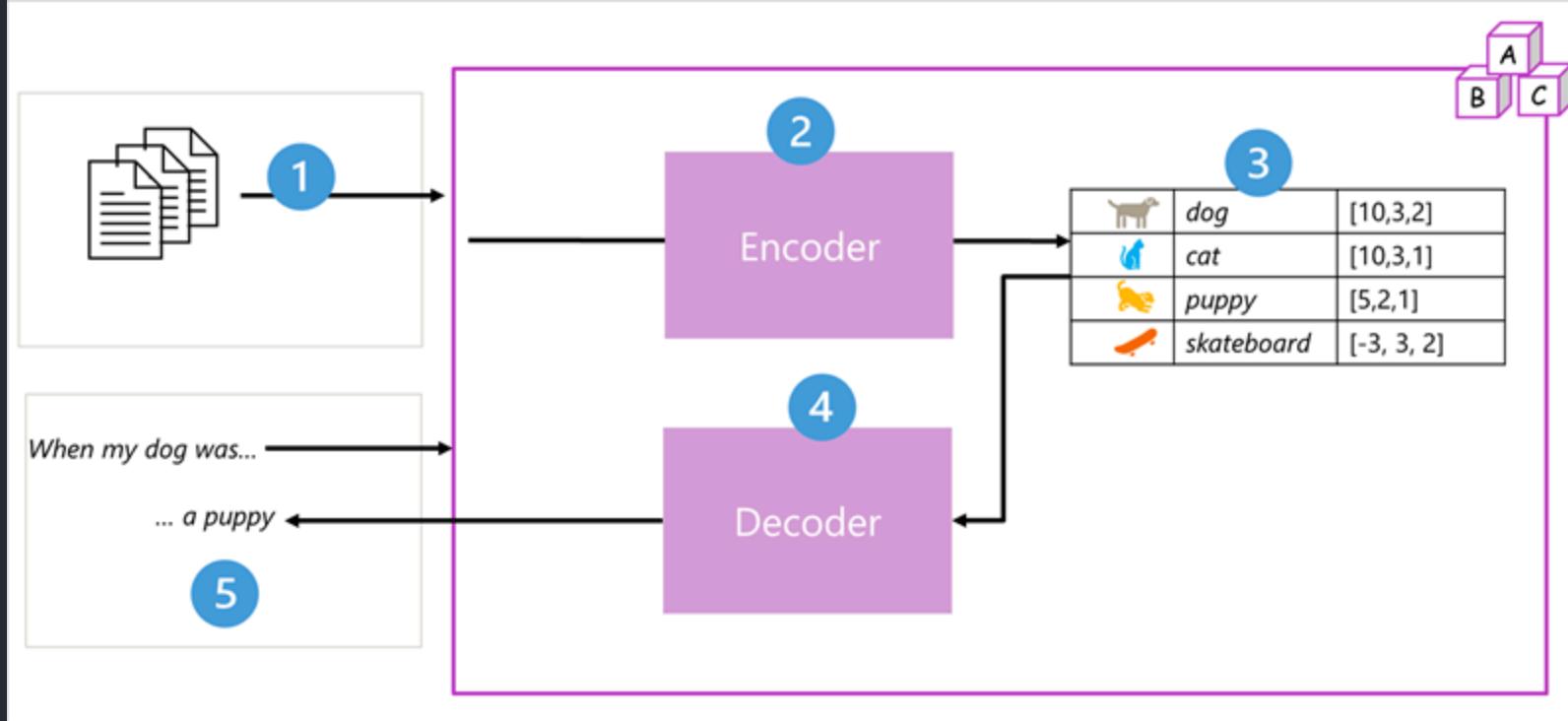
While the mathematical principles behind these language models can be complex, a basic understanding of the architecture used to implement them can help you gain a conceptual understanding of how they work.

Transformer models

Machine learning models for natural language processing have evolved over many years. Today's cutting-edge large language models are based on the *transformer* architecture, which builds on and extends some techniques that have been proven successful in modeling *vocabularies* to support NLP tasks - and in particular in generating language. Transformer models are trained with large volumes of text, enabling them to represent the semantic relationships between words and use those relationships to determine probable sequences of text that make sense. Transformer models with a large enough vocabulary are capable of generating language responses that are tough to distinguish from human responses.

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.



1. The model is trained with a large volume of natural language text, often sourced from the internet or other public sources of text.
2. The sequences of text are broken down into tokens (for example, individual words) and the encoder block processes these token sequences using a technique called attention to determine relationships between tokens (for example, which tokens influence the presence of other tokens in a sequence, different tokens that are commonly used in the same context, and so on.)
3. The output from the encoder is a collection of vectors (multi-valued numeric arrays) in which each element of the vector represents a semantic attribute of the tokens. These vectors are referred to as embeddings.
4. The decoder block works on a new sequence of text tokens and uses the embeddings generated by the encoder to generate an appropriate natural language output.
5. For example, given an input sequence like "When my dog was", the model can use the attention technique to analyze the input tokens and the semantic attributes encoded in the embeddings to predict an appropriate completion of the sentence, such as "a puppy."

In practice, the specific implementations of the architecture vary – for example, the Bidirectional Encoder Representations from Transformers (BERT) model developed by Google to support their search engine uses only the encoder block, while the Generative Pretrained Transformer (GPT) model developed by OpenAI uses only the decoder block.

While a complete explanation of every aspect of transformer models is beyond the scope of this module, an explanation of some of the key elements in a transformer can help you get a sense for how they support generative AI.

Tokenization

The first step in training a transformer model is to decompose the training text into *tokens* - in other words, identify each unique text value. For the sake of simplicity, you can think of each distinct word in the training text as a token (though in reality, tokens can be generated for partial words, or combinations of words and punctuation).

For example, consider the following sentence:

I heard a dog bark loudly at a cat

To tokenize this text, you can identify each discrete word and assign token IDs to them. For example:

Copy

- I (1)
- heard (2)
- a (3)
- dog (4)
- bark (5)
- loudly (6)
- at (7)
- *(“a” is already tokenized as 3)*
- cat (8)

The sentence can now be represented with the tokens: {1 2 3 4 5 6 7 3 8}. Similarly, the sentence "I heard a cat" could be represented as {1 2 3 8}.

As you continue to train the model, each new token in the training text is added to the vocabulary with appropriate token IDs:

- meow (9)
- skateboard (10)
- *and so on...*

With a sufficiently large set of training text, a vocabulary of many thousands of tokens could be compiled.

Embeddings

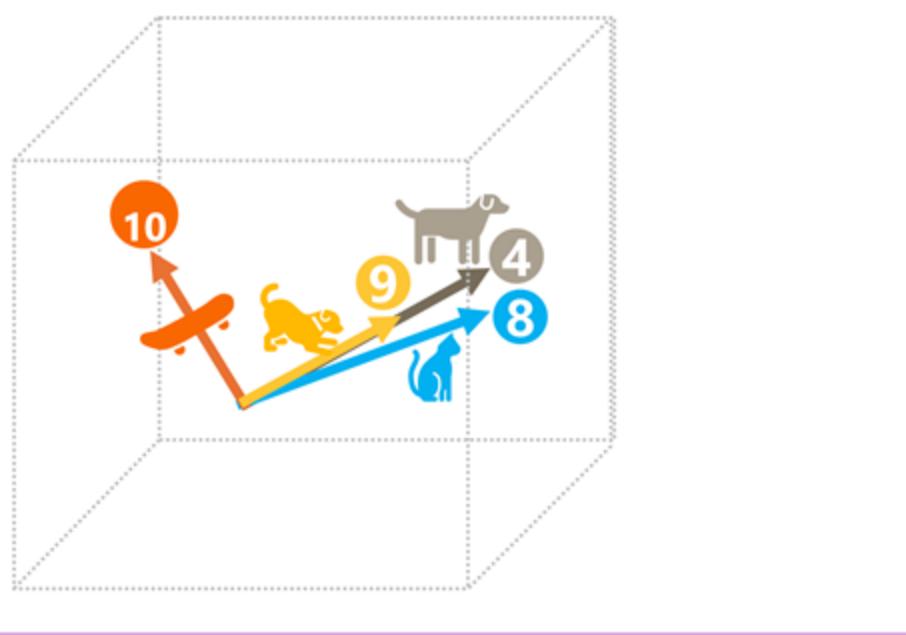
While it may be convenient to represent tokens as simple IDs - essentially creating an index for all the words in the vocabulary, they don't tell us anything about the meaning of the words, or the relationships between them. To create a vocabulary that encapsulates semantic relationships between the tokens, we define contextual vectors, known as *embeddings*, for them. Vectors are multi-valued numeric representations of information, for example [10, 3, 1] in which each numeric element represents a particular attribute of the information. For language tokens, each element of a token's vector represents some semantic attribute of the token. The specific categories for the elements of the vectors in a language model are determined during training based on how commonly words are used together or in similar contexts.

Vectors represent lines in multidimensional space, describing *direction* and *distance* along multiple axes (you can impress your mathematician friends by calling these *amplitude* and *magnitude*). It can be useful to think of the elements in an embedding vector for a token as representing steps along a path in multidimensional space. For example, a vector with three elements represents a path in 3-dimensional space in which the element values indicate the units traveled forward/back, left/right, and up/down. Overall, the vector describes the direction and distance of the path from origin to end.

The elements of the tokens in the embeddings space each represent some semantic attribute of the token, so that semantically similar tokens should result in vectors that have a similar orientation – in other words they point in the same direction. A technique called *cosine similarity* is used to determine if two vectors have similar directions (regardless of distance), and therefore represent semantically linked words. As a simple example, suppose the embeddings for our tokens consist of vectors with three elements, for example:

- 4 ("dog"): [10,3,2]
- 8 ("cat"): [10,3,1]
- 9 ("puppy"): [5,2,1]
- 10 ("skateboard"): [-3,3,2]

We can plot these vectors in three-dimensional space, like this:



The embedding vectors for “dog” and “puppy” describe a path along an almost identical direction, which is also fairly similar to the direction for “cat”. The embedding vector for “skateboard” however describes journey in a very different direction.

Note

The previous example shows a simple example model in which each embedding has only three dimensions. Real language models have many more dimensions.

There are multiple ways you can calculate appropriate embeddings for a given set of tokens, including language modeling algorithms like [Word2Vec](#) or the [encoder](#) block in a transformer model.

Attention

The [encoder](#) and [decoder](#) blocks in a transformer model include multiple layers that form the neural network for the model. We don't need to go into the details of all these layers, but it's useful to consider one of the types of layers that is used in both blocks: [attention](#) layers. Attention is a technique used to examine a sequence of text tokens and try to quantify the strength of the relationships between them. In particular, [self-attention](#) involves considering how other tokens around one particular token influence that token's meaning.

In an encoder block, each token is carefully examined in context, and an appropriate encoding is determined for its vector embedding. The vector values are based on the relationship between the token and other tokens with which it frequently appears. This contextualized approach means that the same word might have multiple embeddings depending on the context in which it's used - for example "the bark of a tree" means something different to "I heard a dog bark."

In a decoder block, attention layers are used to predict the next token in a sequence. For each token generated, the model has an attention layer that takes into account the sequence of tokens up to that point. The model considers which of the tokens are the most influential when considering what the next token should be. For example, given the sequence "I heard a dog," the attention layer might assign greater weight to the tokens "heard" and "dog" when considering the next word in the sequence:

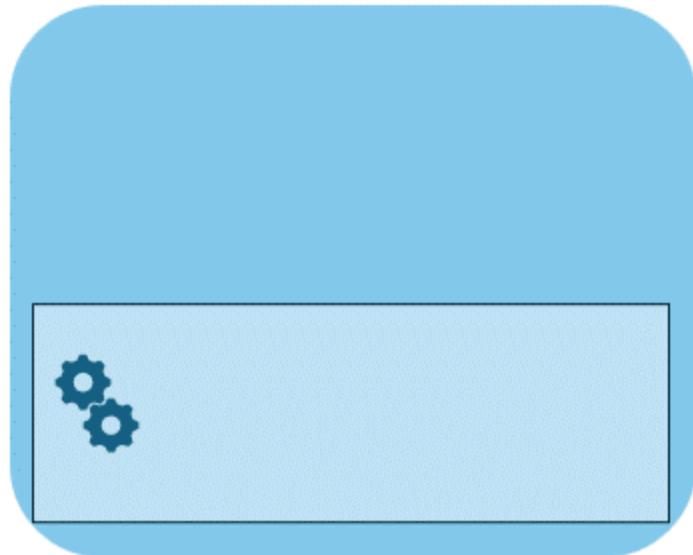
I **heard** a **dog** [bark]

Remember that the attention layer is working with numeric vector representations of the tokens, not the actual text. In a decoder, the process starts with a sequence of token embeddings representing the text to be completed. The first thing that happens is that another *positional encoding* layer adds a value to each embedding to indicate its position in the sequence:

- [1,5,6,2] (I)
- [2,9,3,1] (heard)
- [3,1,1,2] (a)
- [4,10,3,2] (dog)

During training, the goal is to predict the vector for the final token in the sequence based on the preceding tokens. The attention layer assigns a numeric *weight* to each token in the sequence so far. It uses that value to perform a calculation on the weighted vectors that produces an *attention score* that can be used to calculate a possible vector for the next token. In practice, a technique called *multi-head attention* uses different elements of the embeddings to calculate multiple attention scores. A neural network is then used to evaluate all possible tokens to determine the most probable token with which to continue the sequence. The process continues iteratively for each token in the sequence, with the output sequence so far being used regressively as the input for the next iteration – essentially building the output one token at a time.

The following animation shows a simplified representation of how this works – in reality, the calculations performed by the attention layer are more complex; but the principles can be simplified as shown:



1. A sequence of token embeddings is fed into the attention layer. Each token is represented as a vector of numeric values.
2. The goal in a decoder is to predict the next token in the sequence, which will also be a vector that aligns to an embedding in the model's vocabulary.
3. The attention layer evaluates the sequence so far and assigns weights to each token to represent their relative influence on the next token.
4. The weights can be used to compute a new vector for the next token with an attention score. Multi-head attention uses different elements in the embeddings to calculate multiple alternative tokens.
5. A fully connected neural network uses the scores in the calculated vectors to predict the most probable token from the entire vocabulary.
6. The predicted output is appended to the sequence so far, which is used as the input for the next iteration.

During training, the actual sequence of tokens is known – we just mask the ones that come later in the sequence than the token position currently being considered. As in any neural network, the predicted value for the token vector is compared to the actual

value of the next vector in the sequence, and the loss is calculated. The weights are then incrementally adjusted to reduce the loss and improve the model. When used for inferencing (predicting a new sequence of tokens), the trained attention layer applies weights that predict the most probable token in the model's vocabulary that is semantically aligned to the sequence so far.

What all of this means, is that a transformer model such as GPT-4 (the model behind ChatGPT and Bing) is designed to take in a text input (called a prompt) and generate a syntactically correct output (called a completion). In effect, the “magic” of the model is that it has the ability to string a coherent sentence together. This ability doesn't imply any “knowledge” or “intelligence” on the part of the model; just a large vocabulary and the ability to generate meaningful sequences of words. What makes a large language model like GPT-4 so powerful however, is the sheer volume of data with which it has been trained (public and licensed data from the Internet) and the complexity of the network. This enables the model to generate completions that are based on the relationships between words in the vocabulary on which the model was trained; often generating output that is indistinguishable from a human response to the same prompt.

Using language models

Completed 100 XP

- 3 minutes

Organizations and developers can train their own language models from scratch, but in most cases it's more practical to use an existing foundation model, and optionally fine-tune it with your own training data. There are many sources of model that you can use.

On Microsoft Azure, you can find foundation models in the Azure OpenAI service and in the Model Catalog. The Model Catalog is a curated source of models for data scientists and developers using Azure AI Studio and Azure Machine Learning. This offers the benefit of cutting-edge language models like the generative pre-trained transformer (GPT) collection of models (on which ChatGPT and Microsoft's own generative AI services are based) as well as the DALL-E model for image generation. Using these models from the Azure OpenAI service means that you also get the benefit of a secure, scalable Azure cloud platform in which the models are hosted.

In addition to the Azure OpenAI models, the model catalog includes the latest open-source models from Microsoft and multiple partners, including:

- OpenAI
- HuggingFace
- Mistral
- Meta and others.

Large and small language models

There are many language models available that you can use to power generative AI applications. In general, language models can be considered in two categories: *Large Language Models* (LLMs) and *Small Language models* (SLMs).

Expand table

Large Language Models (LLMs)	Small Language Models (SLMs)
LLMs are trained with vast quantities of text that represents a wide range of general subject matter – typically by sourcing data from the Internet and other generally available publications.	SLMs are trained with smaller, more subject-focused datasets
When trained, LLMs have many billions (even trillions) of parameters (weights that can be applied to vector embeddings to calculate predicted token sequences).	Typically have fewer parameters than LLMs.
Able to exhibit comprehensive language generation capabilities in a wide range of conversational contexts.	This focused vocabulary makes them very effective in specific conversational topics, but less effective at more general language generation.
Their large size can impact their performance and make them difficult to deploy locally on devices and computers.	The smaller size of SLMs can provide more options for deployment, including local deployment to devices and on-premises computers; and makes them faster and easier to fine-tune.
Fine-tuning the model with additional data to customize its subject expertise can be time-consuming, and expensive in terms of the compute power required to perform the additional training.	Fine-tuning can potentially be less time-consuming and expensive.

What are copilots?

Completed 100 XP

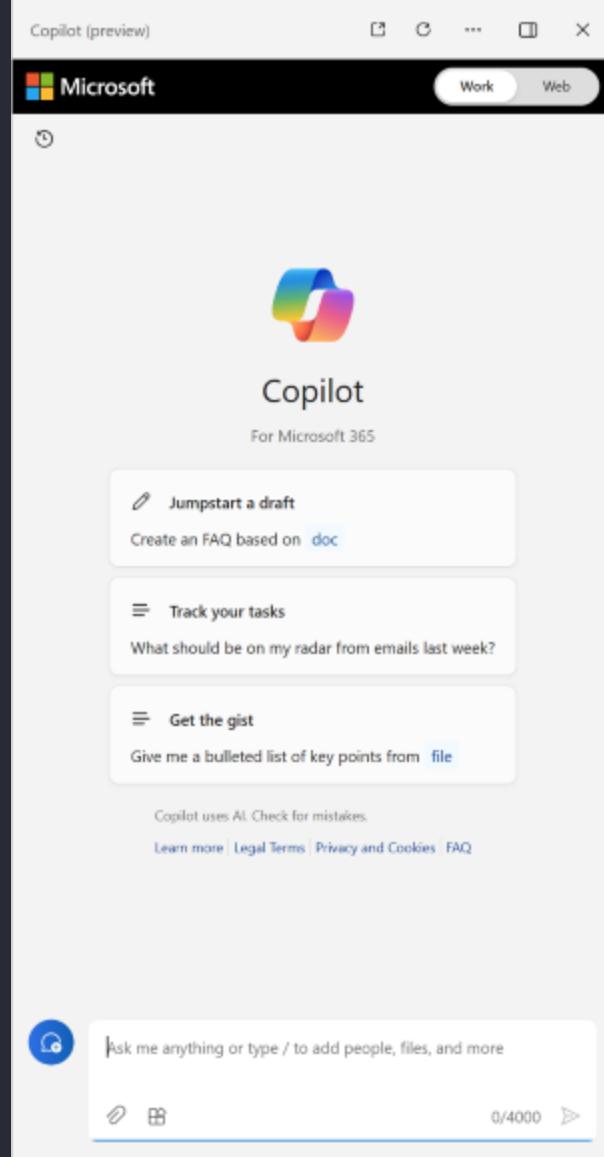
- 3 minutes

The availability of language models has led to the emergence of new ways to interact with applications and systems through digital *copilots*. Copilots are generative AI assistants that are integrated into applications often as chat interfaces. They provide contextualized support for common tasks in those applications.

Microsoft Copilot is integrated into a wide range of Microsoft applications and user experiences. It is based on an open architecture that enables third-party developers to create their own plug-ins to extend or customize the user experience with Microsoft Copilot. Additionally, third-party developers can create their own copilots using the same open architecture.

Business users can use copilots to boost their productivity and creativity with AI-generated content and automation of tasks. Developers can extend copilots by creating plug-ins that integrate them into business processes and data, or even create custom copilots to build generative AI capabilities into apps and services.

Copilots have the potential to revolutionize the way we work by helping with first drafts, information synthesis, strategic planning, and much more. The goal of copilot features is to empower people to be smarter, more productive, more creative, and connected to the people and things around them.



Levels of Copilot adoption

In general, you can categorize industry and personal Copilot adoption into three buckets: off-the-shelf use, extending Microsoft Copilot, and custom development.

- You can use off-the-shelf copilots, like Microsoft Copilot for Microsoft 365 to empower users and increase their productivity.
- You can extend Microsoft Copilot to support custom business processes or tasks, using your own data to control how Copilot responds to user prompts in your organization.

- You can build custom copilots to integrate generative AI into business apps or to create unique experiences for your customers.
-

Microsoft Copilot

Completed 100 XP

- 6 minutes

Microsoft Copilot features can be found throughout all different Microsoft applications. They unlock productivity across your organization, safeguard your business, and build and extend your AI capabilities. Explore some of the different use cases for Microsoft Copilot below.

Web browsing with AI

Microsoft Copilot: use Microsoft Copilot to answer questions, create content, and search the web with the Microsoft Copilot app at <https://copilot.microsoft.com>, when using the Bing search engine, and in the Edge browser. For example, you can ask Microsoft Copilot to generate a list of opportunities in an industry or give more detailed information from your search results.

Copilot

https://copilot.microsoft.com

COPilot Work NOTEBOOK Sign in

Copilot

Your everyday AI companion

green piece of software that
a virtual tree for every bug

Create a table that analyzes public
funding for the arts compared to
GDP

I'm thinking of making a career
change. Can you help me go
through pros and cons?

Design i
repair g

Ask me anything...

0/4000 ➤

When you browse with Microsoft Edge, Copilot is built right in. You can open the Copilot pane in the browser and use it to research topics and create new content – for example to publish a blog post. With all of these Copilot options, signing in with a work or school account enables you to use Copilot in the context of your organization's data and services – enabling you to get assistance with internal resources and information.

Artificial Intelligence Solutions | +

https://www.microsoft.com/en-us/ai?rtc=1

Microsoft

Microsoft AI

12:15 You Create a realistic image of a happy dog singing.

Copilot

Powered by DALL-E 3

Ask me anything...   

12:15 You What kind of flower is this?

Copilot

Based on the image you sent, I think the flower is a hibiscus. Hibiscus is a genus of tropical plants that produce large, showy flowers with prominent stamens. The flowers can be pink, red, yellow, orange, or white, and they often have a contrasting eye in the center.

Ask me anything...   

12:15 You Write an email to congratulate my colleague on her promotion.

Copilot

Sure, I can help you with writing an email to congratulate your colleague on her promotion. Here is a possible email template that you can use:

Subject: Congratulations on your promotion!

Hi [colleague's name],

I am so happy to hear the news of your promotion to [new position]. You deserve this recognition for your hard work and remarkable skills.

Ask me anything...   

MICROSOFT AI

You dream it. AI helps build it.

Explore how Microsoft is empowering the world to achieve more with AI.

Watch video

Chat Compose

Copilot Your everyday AI companion

Hi, see what's possible with Copilot in Edge

Generate page summary

Suggest questions about this page

Choose a conversation style

More Creative More Balanced More Precise

Sign in to ask more questions and have longer conversations

Using relevant sources

Ask me anything or type "@" 

  0/2000 

AI assistance for information workers

Copilot for Microsoft 365: Microsoft Copilot for Microsoft 365 integrates Copilot into the productivity applications that information workers use every day. You can use Copilot in Microsoft Word to generate a new document based on a natural language prompt, and then refine, summarize, and improve the document with a few prompts.

The screenshot shows the Microsoft Word interface with the Copilot ribbon tab selected. A draft document titled "Draft with Copilot" is open, containing the text: "Write a document summarizing considerations for an organization that is planning to adopt Microsoft Copilot for enterprise users." Below the text are two buttons: "Generate" and "Reference a file".

The main content area displays the title "Microsoft Copilot for Enterprise Users: Key Considerations" and a subtitle "A brief guide for organizations that want to leverage the power of AI-assisted coding".

The "What is Microsoft Copilot?" section lists the following bullet points:

- Microsoft Copilot is a cloud-based AI service that helps developers write code faster and smarter.
- It uses machine learning models trained on billions of lines of code to suggest relevant and high-quality code completions, snippets, and entire functions.
- It works with a variety of programming languages, frameworks, and domains, and can adapt to the developer's style and context.
- It is integrated with GitHub Codespaces and Visual Studio Code, and can be accessed through a browser extension or a desktop app.

The Copilot sidebar on the right side of the screen provides options to "Summarize this doc" and "Ask a question about this doc...". It also includes a message input field with placeholder text "Ask me anything about this document" and a word count indicator "0/2000".

You can use Copilot in Microsoft PowerPoint to create a whole presentation based on the contents of a document or email, and then add graphics, reformat slides, and otherwise improve your presentation.

The screenshot shows a Microsoft PowerPoint presentation titled "Introduction to Microsoft Copilot". The slide content is as follows:

Introduction to Microsoft Copilot

This presentation was automatically generated by PowerPoint Copilot based on content found in this document.

The Copilot ribbon is open, showing the following sections:

- Record
- Present in Teams
- Share
- Clipboard
- New Slide
- Reuse Slides
- Slides
- Font
- Paragraph
- Drawing
- Editing
- Dictate
- Voice
- Sensitivity
- Add-ins
- Designer
- Copilot

The Copilot panel displays the following content:

- Create presentation from file https://microsoft-my.sharepoint.com/:w/p/_/Ebu4820BuYJkqo1VD10_BcsBoNuWmQw1-EuGm0JtbPZ3fQ?e=Nausjt
- Here's the outline for the presentation I'm generating:
 - Introduction to Microsoft Copilot
 - Understanding Microsoft Copilot
 - Benefits for Enterprise Users
 - Adoption Strategy for Enterprises
- AI-generated content may be incorrect (like) (dislike)
- Here's a draft of Copilot.docx as a presentation.

I've drafted 5 slides and made some changes based on your doc:

Included example notes containing

In Microsoft Outlook, Copilot can help you summarize your emails, check your schedule, and even find relevant emails and documents to prepare for meetings.

These are just some examples of how you can use Microsoft Copilot for Microsoft 365. There's lots more you can accomplish in Windows, Excel, Teams, and other apps. Learn more at <https://www.microsoft.com/microsoft-365/enterprise/copilot-for-microsoft-365>

Home View Help

New mail



Delete

Archive



Chat



Share to Teams



Flag / Unflag



Pin / Unp

Inbox

William Carter

Copilot adoption meeting prep

12:54

Hey! Can you please send me any relevant pre-r...

Copilot adoption meeting prep General

Summary by Copilot

You received an email from Bill requesting pre-read materials

Bill wants to ensure their team is prepared for the meeting an

Bill is looking forward to the discussion.

AI-generated content may be incorrect



William Carter

To: Emily Lopez



Hey!

Can you please send me any relevant pre-read materi
Copilot adoption?

I want to make sure my team come to the meeting pre
relevant questions they want to ask.

Looking forward to the discussion!

Copilot adoption meeting ...

(No subject)



When am I meeting with
William about Copilot
Adoption?

Your next meeting with William Carter
about Copilot Adoption is on **Saturday,
May 4th, 2024 at 01:00 PM**. It is a
Meeting about Copilot organized by
Emily Lopez. The attendees are
William Carter and **Emily Lopez**.

The meeting objectives include
discussing Copilot adoption. 1 Is there
anything else I can help you with?

Copy

AI-generated content may be incorrect



1 Reference

Try using one of the prompts above

0/2000



Use AI to support business processes

Copilot in Dynamics 365 Customer Service: Modernizes contact centers with generative AI. Customer service agents use Copilot to analyze support tickets, research similar issues, find resolutions, and communicate them to users with only a few clicks and prompts.

D365 Customer Service Workspace

The screenshot shows the Dynamics 365 Customer Service workspace. On the left, there's a chat window with Diane Plank. The messages include:

- Issue description: Conditional policy error in Visual Studio when user tries to login.
- Is this helpful? Yes No
- You have joined the conversation, 10:10 AM.
- Hello Diane, thanks for reaching out to Azure Support, my name is ivan and I'm the support engineer who will be helping you today. 9:46 AM
- I'll go ahead and look into your conditional policy issue. 10:10 AM
- Would you be able to share details including the error message, your visual studio version and any recent changes to your access policies or environment? 10:11 AM
- Sure, they are all captured in the error log. We are seeing the following: { Error Code: 13003, Request Id: asdasd3243-xdadads3-32332f, Correlation Id: d34232vfd23-dasd23casds ... } 10:12 AM
- Thanks, I'll go ahead and look into it. 10:12 AM

On the right, there's a Copilot (preview) interface showing a timeline of interactions with Diane Plank, including an email from her about Visual Studio issues and a live chat from Ivan Lewis. The Copilot interface also displays suggested responses, keywords from the customer, and a review response section with three options:

- Exclude the user from the policy: If the user is unable to access the app, the customer can choose to exclude the user from the policy. This option should be used sparingly, only in situations where the user is trusted. The user should be added back into the policy or group as soon as possible.
- Disable the policy: Disabling the policy makes sure it doesn't apply when a user tries to sign in. The customer can always come back and enable the policy when they would like to use it.
- Delete the policy: If the policy is no longer required, the customer can delete it.

Suggested response to identified question
 ⚠️ This suggestion is AI generated. Review and edit for accuracy before sending. This feature is in preview.

Keywords from customer

User unable to log in due to a conditional access policy failure with error code 13003 for Visual Studio

Review response

I found three options you can consider:

- Exclude the user from the policy: If the user is unable to access the app, the customer can choose to exclude the user from the policy. This option should be used sparingly, only in situations where the user is trusted. The user should be added back into the policy or group as soon as possible.
- Disable the policy: Disabling the policy makes sure it doesn't apply when a user tries to sign in. The customer can always come back and enable the policy when they would like to use it.
- Delete the policy: If the policy is no longer required, the customer can delete it.

Copilot (preview) Clear chat

Copilot for Dynamics 365 Sales: Sales professionals can use Copilot to quickly find relevant customer and industry information by integrating with the company's customer relationship management (CRM) database and beyond. This can enable an account manager to quickly review and qualify a lead, generate a proposal, and set up a customer engagement to close the deal.

Maya Robinson - Save
Lead · Lead

Lead sales process
Active for 4 days

Other Lead source | Warm Rating | New Status | Serena Davis Owner

Summary Details Files Related ...

Contact

Topic * 10 Eco Espresso Machines

First name Maya

Last name * Robinson

Job title ---

City ---

Mobile phone ---

Email ---

Timeline + :

Enter a note...

Created: 5/11/2023 10:00 AM
Email Status
From: Jane Cooper

Up next

Introduction call
Conference outreach | Step 1 | Today
Share product details and check availability for a meeting.

Call Mark complete ...

Lead score 95 Grade A Improving

Intent to buy went up by 10%
Attended recent events
Budget is lower than average

Details

Assistant

Notifications

Sales Copilot PREVIEW

Chat Compose
An updated lead summary with info from Maya Robinson and Fabrikam is available.
Get updated lead summary

Summarize lead

Lead summary

- Maya Robinson is an existing contact from account Fabrikam
- Maya Robinson's role is Operations Executive
- Maya had been in this role at Fabrikam for 6 years
- Top 3 selling products are CoffeeXL, Airpot Lite, and Freshbrew Pods
- We've closed 3 opportunities with Fabrikam in the last 6 months

Suggested updates

- Job title: Operations executive
- Contact information: 842-123-2241
- City: New York
- Email: m.robinson@fabrikam.com

Update all

AI-generated content may be incorrect

Get latest news for account

Find relevant Sales Navigator profiles

Type / for suggestions

Copilot for Dynamics 365 Supply Chain: Handles changes to purchase orders at scale and assess the impact and risk to help optimize procurement decisions. For example, Copilot identifies the level of impact that changes to purchase orders have on downstream processes and gives advice for next steps.

Supply chain management

https://financeoperations.cloudax.int.dynamics.com/?cmp=demf&mi=dashboard=Dynamics%20365%20Project%20Operations&lang=en-us&ls=Ans_Bing

Dynamics 365 Supply Chain Management

Search for a page

Show all changed orders Options

Copilot Preview

USMF

Confirmed purchase orders with changes

Summary

Low impact changes: 342

There are 108 order changes with low downstream impact, comprising of 342 lines. Delivery date delayed affects 331 lines and 23 items, delivery date rushed affects 10 lines and 8 items, quantity decreased affects 2 lines and 2 items, and quantity increased affects 1 line and 1 item.

Show AI generated

High impact changes: 20

There are 6 order changes with high downstream impact, affecting a total of 20 lines. Delivery date delayed affects 20 lines and affects items A0001, J0007, M003, M004, M007, M008, M0017, M0018, M0019, M0020, M0021, M0030, T0004, T0005. Quantity decreased affects 1 line and affects item M0004.

Show AI generated

Impacted downstream orders: 37

There are 37 orders impacted downstream, includ

Purchase order changes with high risk for downstream impact (6)

Confirm purchase orders Send email Send Teams message

Purchase order	Vendor account	Vendor name	Delivery date	Contact ID
0000328	CN-001	Contoso Asia	6/9/2023	
0000332	US-101	Fabrikam Electronics	6/21/2023	
0000335	JP-001	Contoso Chemicals Japan	6/18/2023	
0000365	US-104	Fabrikam Supplier	6/21/2023	
0000397	CN-001	Contoso Asia	6/21/2023	
000016	US-101	Fabrikam Electronics	6/9/2023	000006

Item number	Product name	Original quantity	Changed quantity	Original delivery date	Current delivery date	Reference
T0005	TelevisionHDTVX59052	350.00		6/8/2023	6/9/2023	Sales order
M0003	Tweeter Speaker Unit	96.00		6/8/2023	6/9/2023	Sales order
M0020	Carbon Dioxide	123.00		6/8/2023	6/9/2023	Sales order
A0001	HDMI 6' Cables	10.00		6/8/2023	6/9/2023	Sales order

June 8, 8:55 AM

Plan with alternative product variants

For the selected purchase order change, this could be an alternative product variant:

- Item number 3983389
- Minimum quantity 500
- Latest delivery date 09/10/2023

AI-generated content may be incorrect

Ask a work question or make a request

AI assisted data analytics

Copilot in Microsoft Fabric: Copilot enables analysts to automatically generate the code they need to analyze, manipulate, and visualize data in Spark notebooks.

Microsoft Notebook 1 - Synapse Data Engine

https://msit.powerbi.com/groups/32a98b35-9076-4d79-8a71-b857b363...

Home Edit Run View

Editing Comments Share

Standard session PySpark (Python) Workspace default

PySpark (Python)

chat-magics generated the following cell. Tokens: 13

Remember that AI can make mistakes, so carefully review code before executing.

df.orderBy(df.ListPrice.desc()).show(10)

6 sec - Command executed in 5 sec 527 ms by Graeme Malc PySpark (Python)

Spark jobs (6 of 6 succeeded) Resources Log

ProductID	ProductName	Category	ListPrice
751	Road-150 Red, 48	Road Bikes	3578.27
749	Road-150 Red, 62	Road Bikes	3578.27
750	Road-150 Red, 44	Road Bikes	3578.27
752	Road-150 Red, 52	Road Bikes	3578.27
753	Road-150 Red, 56	Road Bikes	3578.27
771	Mountain-100 Silv...	Mountain Bikes	3399.99
772	Mountain-100 Silv...	Mountain Bikes	3399.99

Copilot Preview

Let's work with your data

Choose an option below or type a request to get started.

Load data from my lakehouse into a DataFrame

Generate insights from data

Suggest data visualizations

What are the unique products in the DataFrame?

Remove any duplicate rows in the DataFrame.

Type a request or click the sparkle icon for ideas.

AI-generated content can have mistakes. Make sure it's accurate and appropriate before using it. Read preview terms

Session ready AutoSave: On Selected Cell 5 of 5 cells

Copilot in Power BI: When creating Power BI reports, Copilot can analyze your data and then suggest and create appropriate data visualizations from it.



Home



Create



Browse



Workspaces



fabric-ws



Untitled report



Notebook 1



fabric_lake



fabric_lake

...



Power BI

File ▾

View ▾

Reading view

Mobile layout

Open data model

Copilot

Data

Visualizations

Filters



product prices across different categories.

+ Create



Top Selling Categories

Product Category Breakdown

Price Category Comparison

Create a page to analyze the distribution of product prices across different categories.

Created a Product Price Distribution page.

Undo

Describe the report you want to create or ask a question

Content created by AI may not be accurate or appropriate, so review it carefully. [Read terms](#)

Manage IT infrastructure and security

Microsoft Copilot for Security: Provides assistance for security professionals as they assess, mitigate, and respond to security threats.

- [Home](#)
- [Incidents & alerts](#)
 - [Incidents](#)
 - [Alerts](#)
- [Hunting](#)
 - [Advanced hunting](#)
 - [Custom detection rules](#)
- [Actions & submissions](#)
- [Threat intelligence](#)
- [Learning hub](#)
- [Trials](#)
- [Partner catalog](#)
- [Exposure management](#)
 - [Overview](#)
 - [Attack surface](#)
 - [Exposure insights](#)
 - [Secure score](#)
- [Assets](#)
 - [Devices](#)

Advanced hunting

[Help resources](#)[New query](#) | [Get summarized failed logins](#) | [Get logon attempts by domain accounts](#)[Run query](#)

Last 7 days

[Save](#)[Share link](#)[Security Copilot](#)

Query

Query results are presented in your local time zone as per settings. Kusto filters, however, work in UTC.

```
1 let logonAttempts = DeviceLogonEvents
2 | where ActionType == "LogonAttempted"
3 | project Timestamp, DeviceId, AccountDomain;
4 let credentialTheftEvents = DeviceEvents
5 | where ActionType in ("AsrlsassCredentialTheftAudited", "AsrlsassCred
6 | project Timestamp, DeviceId, InitiatingProcessAccountDomain;
7 logonAttempts
8 | join kind=inner credentialTheftEvents on $left.DeviceId == $right.De
9 | summarize count() by AccountDomain
10 | order by count_desc
11
```

[Getting started](#) [Results](#) [Query history](#)**No results found in the specified time frame.**

Security Copilot

Nov 8, 2023 1:43 PM

get logon attempts by domain accounts targeted by credential theft

Nov 8, 2023 1:43 PM

Here's a query you can use to find what you need:

```
let logonAttempts =
DeviceLogonEvents
| where ActionType ==
"LogonAttempted"
| project Timestamp, DeviceId,
AccountDomain;
```

[Add and run](#)

AI generated. Verify for accuracy.

[Ask a question to generate a query](#)

Microsoft Copilot for Azure: Integrated into the Azure portal to assist infrastructure administrators as they work with Azure cloud services.

Home - Microsoft Azure x

https://portal.azure.com/#home

Microsoft Azure Search resources, services, and docs (G+/)

Copilot

Azure services

Create a resource

Resource groups

Subscriptions

Azure Databricks

All resources

Azure AI services

Azure AI services multi...

Microsoft Entra ID

App Services

More services

Resources

Recent Favorite

Name Type

No resources have been favored

Favorite resources to quickly navigate to them from the home page.

Select resources to favorite

Copilot Preview

Design I want to automate the deployment of Azure Virtual Desktop infrastructure using ARM templates.

Operate List all private endpoints in a subnet ending with 'backend'.

Optimize List all my Storage Accounts by last access time.

Troubleshoot I'm facing an error in the portal with my IoT device.

I want to ...

0 / 500

The screenshot shows the Microsoft Azure portal's home page. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The main area is titled 'Azure services' and contains several tiles: 'Create a resource' (plus icon), 'Resource groups' (cube icon), 'Subscriptions' (key icon), 'Azure Databricks' (red cube icon, highlighted with a white box), 'All resources' (grid icon), 'Azure AI services' (cloud icon), 'Azure AI services multi...' (cloud icon), 'Microsoft Entra ID' (blue icon), 'App Services' (blue icon), and 'More services' (arrow icon). Below this is a section titled 'Resources' with tabs for 'Recent' and 'Favorite' (underlined). It shows a table header for 'Name' and 'Type' with one row below it containing a gray hexagonal icon. A message says 'No resources have been favored' and 'Favorite resources to quickly navigate to them from the home page.' A button labeled 'Select resources to favorite' is at the bottom. To the right, a 'Copilot Preview' sidebar is open with four cards: 'Design' (automating VD deployment), 'Operate' (listing private endpoints), 'Optimize' (listing storage accounts by access time), and 'Troubleshoot' (facing an IoT device error). At the bottom of the sidebar is a text input field 'I want to ...' and a character icon.

AI powered software development

GitHub Copilot: Helps developers maximize their productivity by analyzing and explaining code, adding code documentation, generating new code based on natural language prompts, and more.

The screenshot shows a GitHub repository page for 'mslearn-openai' with the path 'Labfiles / 06-use-own-data / Python'. The 'Code' tab is selected, displaying a Python script named 'main.py'. The script imports os, json, and dotenv, and uses openai to interact with an Azure OpenAI model. A Copilot AI sidebar is open on the right, showing a conversation between 'DeveloperUser' and 'Copilot'. 'DeveloperUser' asks, 'What does this file do?' and 'Copilot' responds with a summary of its functionalities. Below the summary, a bulleted list details the script's purpose, including reading environment variables, initializing an Azure OpenAI client, and prompting the user for a question.

```
import os
import json
from dotenv import load_dotenv

# Add OpenAI import
from openai import AzureOpenAI

def main():

    try:
        # Flag to show citations
        show_citations = False

        # Get configuration settings
        load_dotenv()
        azure_oai_endpoint = os.getenv("AZURE_OAI_ENDPOINT")
        azure_oai_key = os.getenv("AZURE_OAI_KEY")
        azure_oai_deployment = os.getenv("AZURE_OAI_DEPLOYMENT")
        azure_search_endpoint = os.getenv("AZURE_SEARCH_ENDPOINT")
```

DeveloperUser 1 minute ago

What does this file do?

1 reference ▾

Copilot 1 minute ago

This Python script is used to interact with an Azure OpenAI model. Here is a summary of its functionalities:

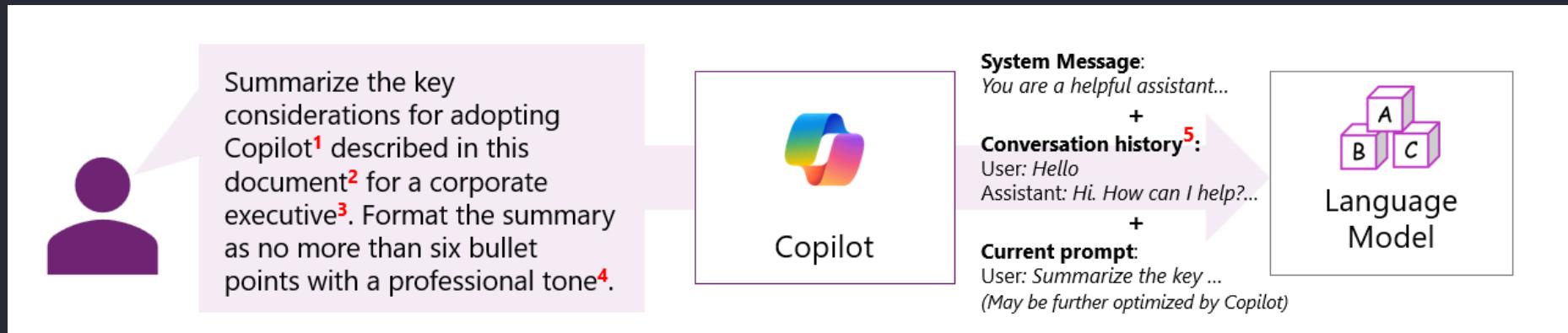
- It imports necessary modules such as `os`, `json`, and `dotenv` for environment variable handling, and `AzureOpenAI` from `openai` for communicating with the Azure OpenAI service.
- In the `main` function:
 - It reads environment variables including the Azure OpenAI endpoint, key, deployment, and Azure Search parameters.
 - Initializes an Azure OpenAI client with the retrieved configuration.
 - Prompts the user to enter a question, which will be

Ask Copilot

Considerations for Copilot prompts

- 3 minutes

The quality of responses from copilots not only depends on the language model used, but on the types of prompts users provide. Prompts are ways we tell an application what we want it to do. You can get the most useful completions by being explicit about the kind of response you want. Take this example, "Summarize the key considerations for adopting Copilot described in this document for a corporate executive. Format the summary as no more than six bullet points with a professional tone." You can achieve better results when you submit clear, specific prompts.



Consider the following ways you can improve the response a copilot provides:

1. Start with a specific goal for what you want the copilot to do
2. Provide a source to ground the response in a specific scope of information
3. Add context to maximize response appropriateness and relevance
4. Set clear expectations for the response
5. Iterate based on previous prompts and responses to refine the result

In most cases, a copilot doesn't just send your prompt as-is to the language model. Usually, your prompt is augmented with:

- A **system message** that sets conditions and constraints for the language model behavior. For example, "You're a helpful assistant that responds in a cheerful, friendly manner." These system messages determine constraints and styles for the model's responses.
- The conversation history for the current session, including past prompts and responses. The history enables you to refine the response iteratively while maintaining the context of the conversation.

- The current prompt – potentially optimized by the copilot to reword it appropriately for the model or to add more grounding data to scope the response.

The term ***prompt engineering*** describes the process of prompt improvement. Both developers who design applications and consumers who use those applications can improve the quality of responses from generative AI by considering prompt engineering.

Extending and developing copilots

Completed 100 XP

- 3 minutes

If your organization makes the decision to customize Microsoft copilot or develop custom copilots, Microsoft provides two tools that you can use, Copilot Studio and Azure AI Studio.

Copilot Studio

Copilot Studio is designed to work well for low-code development scenarios in which technically proficient business users or developers can create conversational AI experiences. The resulting copilot is a fully managed SaaS (software as a service) solution, hosted in your Microsoft 365 environment and delivered through chat channels like Microsoft Teams. With Copilot Studio, the infrastructure considerations and model deployment details are taken care of for you, making it easy to focus on creating an effective solution. For more information, see <https://www.microsoft.com/microsoft-copilot/microsoft-copilot-studio>.

Your copilot is ready! Here's what's next:

- Add actions so your copilot can do things for you
- Build topics to focus and guide how your copilot answers
- Publish your copilot so others can use it

Details

 Name
Expense Claim Assistance

Description
Assists employees with expense claims by providing guidance, support, and explanations related to submitting and managing expense claims.

Instructions
Your name is Expense Claim Assistance. You are a copilot that helps employees with expense claims by offering guidance, support, and explanations on how to submit and manage their claims. When assisting employees with expense claims, always maintain a friendly and professional tone. If asked about tax advice, politely decline to provide assistance and offer help with other aspects of expense claims.

Test your copilot

Hello, I'm Expense Claim Assistance, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.

Just now

Ask a question or describe what you need

0/2000

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

Azure AI Studio

Azure AI Studio is a PaaS (platform as a service) development portal for professional software developers that gives you full control over the language model you want to use, including the capability to fine-tune the model with your own data. You can define prompt flows that orchestrate conversation flow and integrate your own data augmentation and prompt engineering logic, and you can

deploy the resulting copilot service in the cloud and consume it from custom-developed apps and services. For more information, see <https://azure.microsoft.com/products/ai-studio/>.

The screenshot shows the Azure AI Studio interface, specifically the Model catalog page. The URL in the browser is <https://ai.azure.com/explore/models?wsid=/subscriptions/c>. The left sidebar includes sections for Project overview, Settings, Get started, Model catalog (which is selected), Model benchmarks, Prompt catalog, Project playground, Chat, Completion, Assistants (PREVIEW), Images, Tools (Code PREVIEW, Prompt flow, Tracing, Evaluation PREVIEW, Fine-tuning PREVIEW), Components (Data, Indexes, Deployments), and Deployments. The main content area is titled "Find the right model to build your custom AI solution" and displays a list of 1667 models. The first few models listed are gpt-4o (Chat completion), gpt-4 (Chat completion), dall-e-3 (Text to image), gpt-35-turbo-instruct (Chat completion), davinci-002 (Completions), and text-embedding-ada-002 (Embeddings). To the right of the model list are filter panels for Collections (Curated by Azure AI, Azure OpenAI, Meta, Hugging Face, NVIDIA, Microsoft, Mistral AI, Deci AI, Nixtla, JAIS, Cohere, Databricks, Snowflake), Deployment options (Managed compute, Serverless API), Inference tasks (Conversational, Fill mask, Question answering, Summarization), and Fine-tuning tasks (Image classification).

Exercise - Explore Microsoft Copilot

- 40 minutes

Now let's explore generative AI with Microsoft Copilot.

Note

There are two variations of the exercise: one for learners who have access to Microsoft 365 Copilot, and an alternative for learners who don't. **Launch and complete one or the other to explore Microsoft Copilot.**

Option 1: Using Microsoft Copilot for Microsoft 365

If you have access to Microsoft Copilot for Microsoft 365, launch this version of the exercise and follow the instructions.

Note

To complete this version, you need a Microsoft 365 account and a license for Microsoft Copilot.

[Launch Exercise](#)

Option 2: Using Microsoft Copilot in Microsoft Edge

If you do not have access to Microsoft Copilot for Microsoft 365, launch this version to explore Microsoft Copilot in Microsoft Edge.

This exercise requires a free Microsoft account (for example an *outlook.com*, *live.com*, or *hotmail.com* account), and the Microsoft Edge web browser.

1. If you don't already have Microsoft Edge installed, download and install it for free from <https://www.microsoft.com/edge/download>.
2. If you don't already have a Microsoft account, sign up for one at <https://signup.live.com>.
3. If you haven't already, sign into Microsoft Edge using your Microsoft account.

Tip

If you did not sign into Edge after creating your account, use the user account icon (👤) on the Edge toolbar to sign in using a **personal** account.

Summary

Completed 100 XP

- 1 minute

Generative AI is a rapidly developing field of AI that supports new language generation, code development, image creation, and more. This module explored how language models power generative AI and how copilot brings generative AI capabilities to your desktop. You learned the basics of Large Language Models (LLMs) and Small Language Models (SLMs), and the underlying transformer architecture of advanced language models.

The main takeaways from this module include understanding the role of generative AI in creating AI assistants or copilots that provide contextualized support for common tasks. Often, generative AI is integrated into chat applications like Microsoft Copilot to interpret natural language inputs and generate suitable responses. You learned about the three levels of copilot adoption and the various applications of Microsoft Copilot across different Microsoft applications. The module also highlighted the importance of improving the quality of responses from generative AI. Finally, you have been introduced to two tools offered by Microsoft for customizing or developing custom copilots: Copilot Studio and Azure AI Studio.

More reading:

- [Use Microsoft Copilot](#)
- [Learn about Copilot for Microsoft 365](#)
- [Try out Microsoft Copilot Studio](#)
- [Try out Microsoft AI Studio](#)

Fundamentals of Generative AI

You have earned an achievement!

Congratulations, but what should you do next?