

# Using Large Language Models with Python

Python Cloud Advocate at Microsoft

Formerly: UC Berkeley, Khan Academy,  
Woebot, Coursera, Google

Find me online at:

---

Mastodon [@pamelaf@fosstodon.org](https://fosstodon.org/@pamelaf)

---

Twitter [@pamelaf](https://twitter.com/pamelaf)

---

GitHub [www.github.com/pamelaf](https://www.github.com/pamelaf)

---

Website [pamelaf.org](https://pamelaf.org)





## Artificial Intelligence

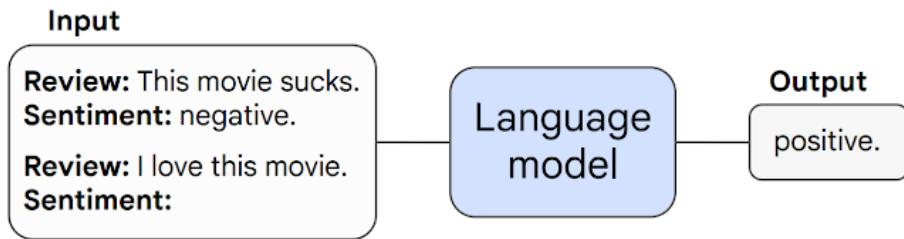
Machine learning

Deep learning

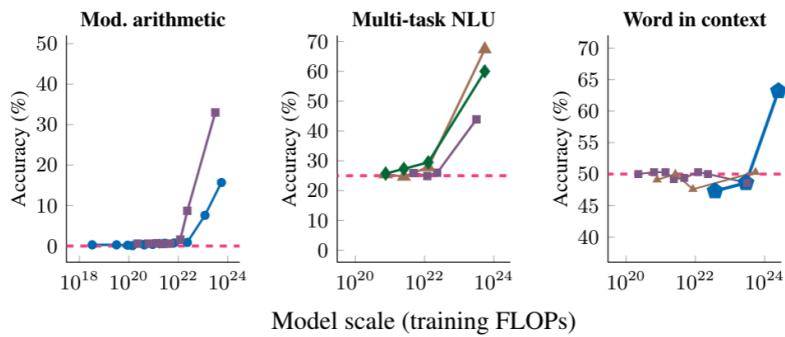
Generative AI

- 1956: **Artificial Intelligence:**  
The field of computer science that seeks to create intelligent machines that can replicate or exceed human intelligence
- 1997: **Machine Learning:**  
Subset of AI that enables machines to learn from existing data and improve upon that data to make decisions or predictions
- 2017: **Deep Learning:**  
A machine learning technique in which layers of neural networks are used to process data and make decisions
- 2021: **Generative AI:**  
Create new written, visual, and auditory content given prompts, often using Large Language Models or Diffusion models

An LLM is a model that is so large that it achieves general-purpose language understanding and generation.



—●— LaMDA   —■— GPT-3   —◆— Gopher   —▲— Chinchilla   —●— PaLM   - - - Random



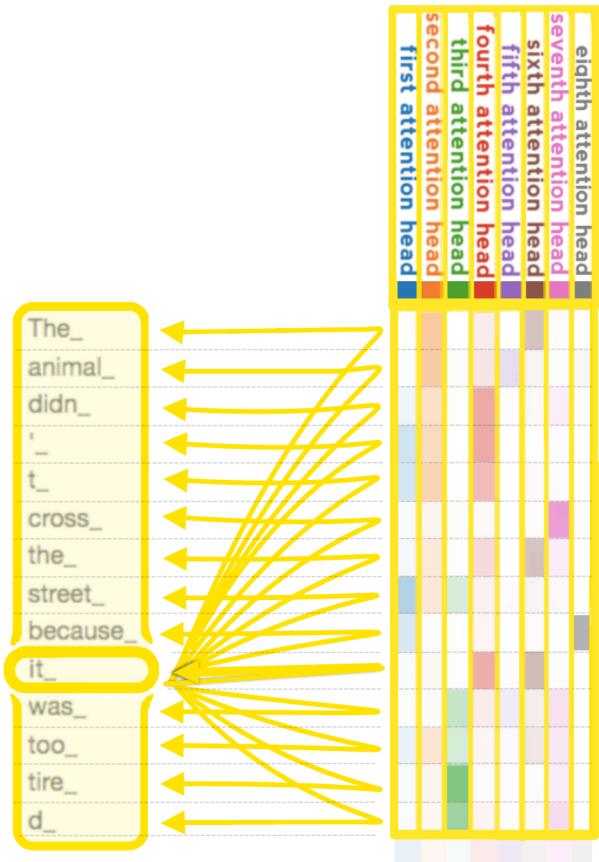
From Characterizing Emergent Phenomena in LLMs

GPT models are LLMs based on  
Transformer architecture from:

 ["Attention is all you need" paper](#)  
by Google Brain

Learn more:

- Andrej Karpathy:  [State of GPT](#)
- Andrej Karpathy:  [Let's build  
GPT: from scratch, in code](#)





pamelafox



Whats a generator in Python?



GitHub Copilot

In Python, a generator is a special type of function that returns an iterator object. Unlike regular functions, which return a value and then exit, generators can pause execution and return a value multiple times, resuming where they left off each time.



You

need lunch ideas. i have carrot pepper tortilla cheese.



ChatGPT

With the ingredients you have available, here are a few lunch ideas:

1. **Vegetable Quesadilla:** Use the carrot pepper tortilla and cheese to make a delicious vegetable quesadilla. You can add some extra veggies like sliced bell peppers, onions, spinach, or mushrooms for added flavor and nutrition. Serve it with salsa, guacamole, or sour cream on the side.
2. **Grilled Cheese Sandwich:** Make a classic grilled cheese sandwich using the carrot pepper tortilla and cheese. You can also add some thinly sliced tomatoes or avocado for extra flavor. Grill the sandwich until the tortilla is crispy and the cheese is melted.



You

how much to pay employee whose normal pay is 25/hour and who works 24 hours extra? (assuming 1.5x overtime rate)



Copilot

To calculate the total pay for an employee with a normal rate of \$25 per hour and 24 hours of overtime at a rate of 1.5 times the normal pay, you can use the following formula:

1. Calculate the regular pay for the standard work hours (assuming a standard 40-hour workweek):

$$\text{Regular Pay} = \text{Regular Hourly Rate} \times \text{Regular Hours}$$

$$\text{Regular Pay} = \$25/\text{hour} \times 40\text{hours} = \$1000$$

2. Calculate the overtime pay for the extra hours:

$$\text{Overtime Pay} = \text{Regular Hourly Rate} \times 1.5 \times \text{Extra Hours}$$

$$\text{Overtime Pay} = \$25/\text{hour} \times 1.5 \times 24\text{hours} = \$900$$

3. Add the regular pay and the overtime pay to get the total pay:

$$\text{Total Pay} = \text{Regular Pay} + \text{Overtime Pay}$$

$$\text{Total Pay} = \$1000 + \$900 = \$1900$$

```
def calculate_days_between(start_iso, end_iso):  
    from datetime import datetime  
    start = datetime.strptime(start_iso, "%Y-%m-%d")  
    end = datetime.strptime(end_iso, "%Y-%m-%d")  
    return (end - start).days
```

```
1 def calculate_days_between(start_iso, end_iso):  
    Add type annotations  
    Copilot generated code may be incorrect
```

ChatGPT, GitHub Copilot, Bing Copilot, and many other tools are powered by LLMs.

Hosted LLMs can only be accessed via API, from a company hosting the model and infrastructure for you.

<b>Company</b>	<b>Model</b>	<b>Parameters</b>
OpenAI	<u>GPT-3.5</u>	175B
OpenAI	<u>GPT-4</u>	Undisclosed
Google	<u>PaLM</u>	540B
Google	<u>Gemini 1, 1.5</u>	Undisclosed
Anthropic	<u>Claude 3 family</u>	Undisclosed

 [OpenAI models overview](#)

## Assistant setup

X

 System message

 Add your data (preview)

 Save changes

### Specify how the chat should act

Use a template to get started, or just start writing your own system message below. Want some tips? [Learn more](#)

#### Use a system message template

Select a template

#### System message

You are an AI assistant that helps people find information.

#### Examples

 Add an example

## Chat session

 Clear chat

 View code

 Show raw JSON

Write a 3-sentence story about a pony that turns into an Alicorn.



Once upon a time, in a magical land filled with vibrant colors and sparkling sunshine, lived a humble pony named Starlight. One day, Starlight discovered an ancient stone that bestowed her with magical powers, causing her to metamorphose into a majestic Alicorn. As an Alicorn, she used her newfound magic and wings to protect her land, bringing peace and harmony to all its inhabitants.

Type user query here. (Shift + Enter for new line)



A local LLM can be downloaded and used by anyone, as long as they have the computational resources to run it.

<b>Company</b>	<b>LLM</b>	<b>Parameters</b>
Meta	<a href="#"><u>Llama 2</u></a>	7b, 13b, 70b
Google	Gemma	2b, 7b
Microsoft research	<a href="#"><u>Phi-2</u></a>	2.7b
Mistral AI	Mistral	7b
Mistral AI	Mixtral	8x7b
Researchers	Llava	7b, 13b, 34b

# Ollama is a tool for easily running local LLMs on your computer.

```
(.venv) ollama-time % ollama run llama2
>>> write a 3-sentence story about pony that turns into alicorn

As the sun set over the magical land of Equestria, Ponyville's very own Twilight Sparkle gazed up at the sky in wonder. With a flutter of her mane and a shimmer of her tail, she transformed into an Alicorn, radiating an ethereal light that illuminated the entire town. Now empowered with the magic of the alicorns, Twilight was ready to face whatever challenges lay ahead, armed with the strength and grace of her new form.

>>> █end a message (/? for help)
```

# in Python



The [OpenAI API](#) is an HTTP API with endpoints for different tasks, like **chat completions** and **embeddings**.

Use with either:

- [Openai.com account](#)
- [Azure OpenAI account](#)
- Local LLM with OpenAI-compatible API  
(Ollama/llamafile)

Once you have access, you can use the API from Python. Get started with:

 [Python OpenAI Demos](#)

 [Ollama Python Playground](#)

If you're not inside a dev container or Jupyter notebook, create a virtual environment:

```
python3 -m venv venv  
source venv/bin/activate
```

Install the OpenAI Python library:

```
pip install openai
```

For openai.com OpenAI, set your API key:

```
client = openai.OpenAI(api_key="your-api-key")
```

For Azure OpenAI, use Azure default credentials:

```
azure_credential = azure.identity.DefaultAzureCredential()
token_provider = get_bearer_token_provider(azure_credential,
    "https://cognitiveservices.azure.com/.default")

client = openai.AzureOpenAI(
    api_version="2024-03-01-preview",
    azure_endpoint=f"https://your-openai-service.openai.azure.com",
    azure_ad_token_provider=token_provider,
)
```

# Configure the client to point at local server:

```
client = openai.OpenAI(  
    base_url="http://localhost:11434/v1",  
    api_key="nokeyneeded",  
)
```

 [Ollama OpenAI compatibility](#)

 [Ollama Python Playground](#)

## Using chat completions API:

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages = [  
        {"role": "system",  
         "content": "You are a helpful assistant.."},  
        {"role": "user",  
         "content": "What can I do on my trip to Tokyo?"}  
    ],  
    max_tokens=400,  
    temperature=1,  
    top_p=0.95)  
  
print(response.choices[0].message.content)
```

## Full example: chat.py

```
completion = client.chat.completions.create(  
    stream=True,  
    messages = [  
        {"role": "system",  
         "content": "You are a helpful assistant.."},  
        {"role": "user",  
         "content": "What can I do on my trip to Tokyo?"}  
    ])  
  
for event in completion:  
    print(event.choices[0].delta.content)
```

## Full example: chat.py

## Pros:

- Creative 😊
- Great with patterns
- Good at syntax (natural and programming)

## Pros:

- Creative 😊
- Great with patterns
- Good at syntax (natural and programming)

## Cons:

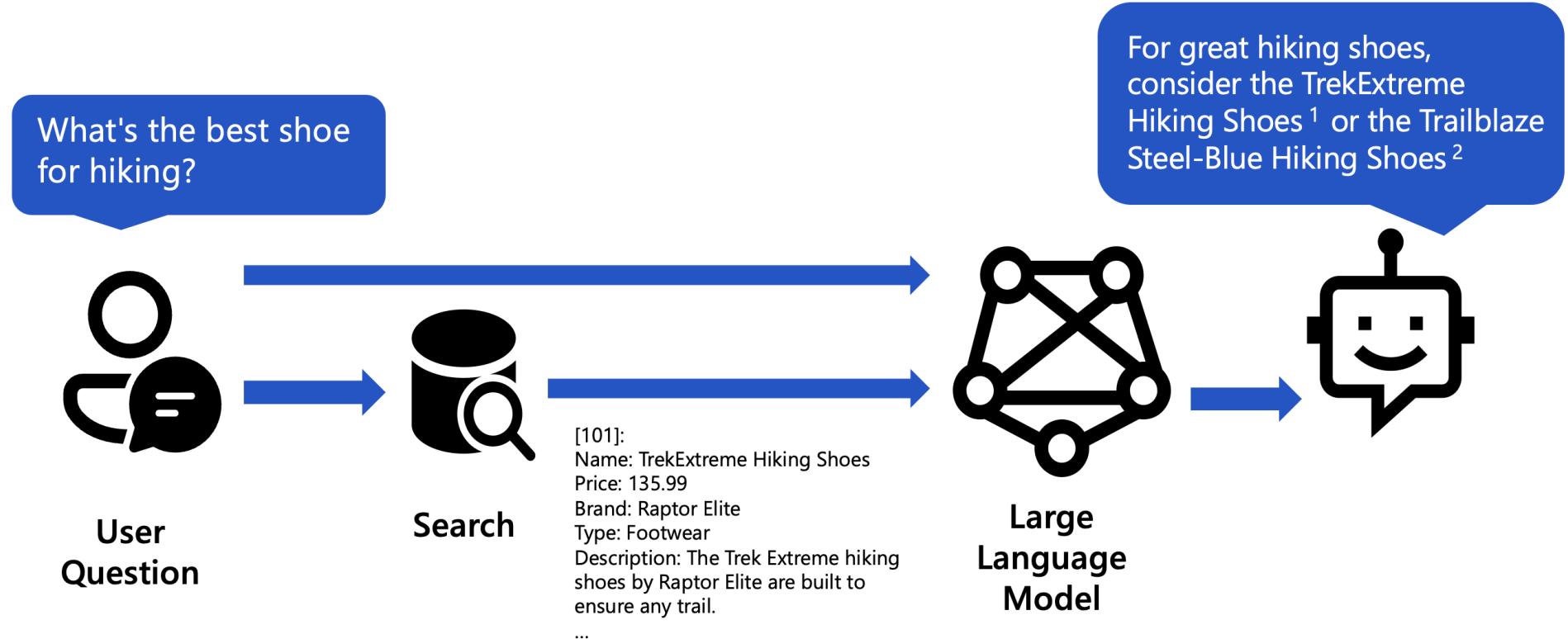
- Creative 🤪
- Makes stuff up (unknowingly)
- Limited context window (4K-32K)

- **Prompt engineering:** Request a specific tone and format
- **Few-shot examples:** Demonstrate desired output format
- **Chained calls:** Get the LLM to reflect, slow down, break it down
- **Retrieval Augmented Generation (RAG):** Supply just-in-time facts
- **Fine tuning:** Teach LLM new facts/syntax by permanently altering weights

# Generation



# Use a retrieval system to find the best context for the generation model.



## **Retrieval system (Search)**

- Organize knowledge to fit needs of models
- Retrieve relevant information
- Ensure data freshness
- Enforce access control

## **→ Generative model (LLM)**

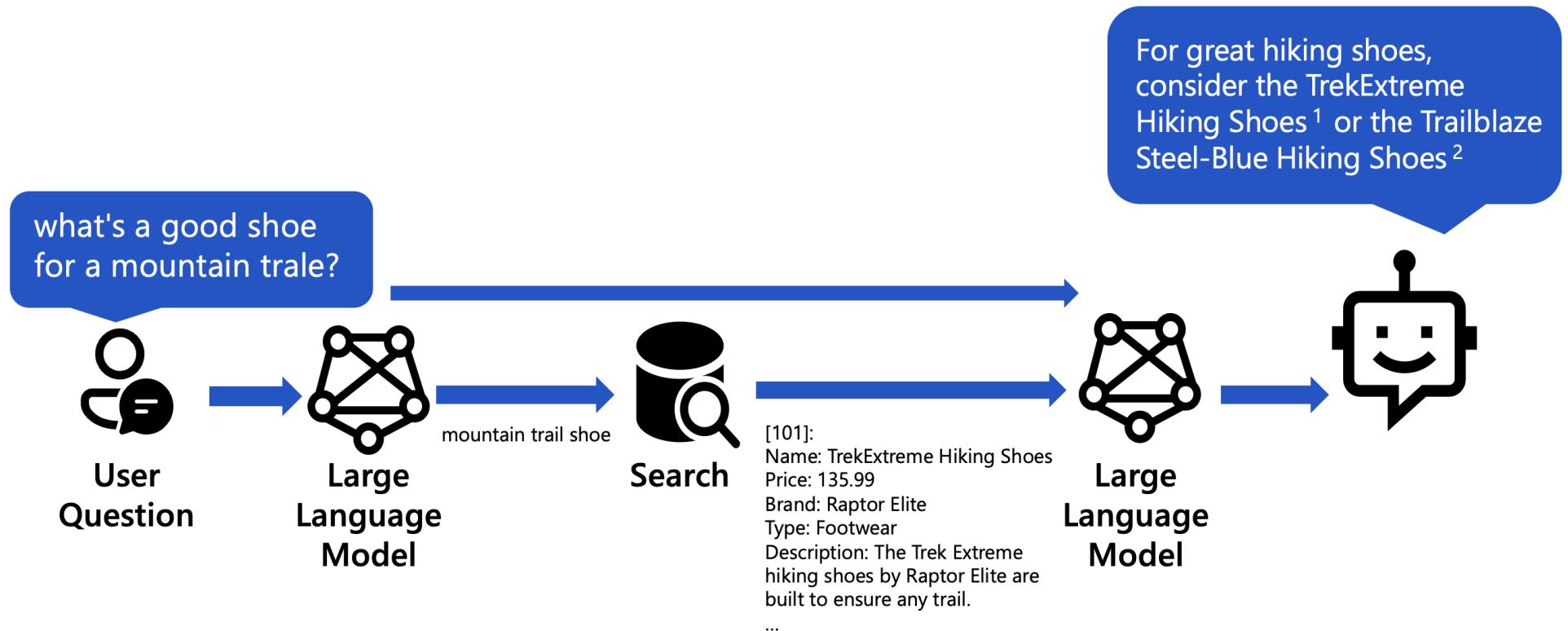
- Summarize information
- Answer questions
- Suggest follow-up questions

[github.com/Azure-Samples/rag-postgres-openai-python/](https://github.com/Azure-Samples/rag-postgres-openai-python/)

The screenshot shows a chat interface with the following elements:

- Header:** "Clear chat" and "Developer settings" buttons.
- Input Field:** "Best shoe for hiking?"
- Response Area:**
  - Icon:** A blue starburst icon with a plus sign.
  - Text:** "For a great hiking shoe option, you can consider the Trailblaze Steel-Blue Hiking Shoes by WildRunner".  
These shoes are rugged and durable, featuring a waterproof membrane, high-traction rubber outsole, and a reinforced toe cap for extra protection. They are designed for superior grip on any terrain and are perfect for hiking, camping, and other outdoor adventures. The steel-blue color adds a stylish touch to your outdoor gear.
  - Section:** "References:"
  - Item 1:** "1. Trailblaze Steel-Blue Hiking Shoes"  
*Brand: WildRunner*  
*Price: 129.99*
  - Text:** "Explore the great outdoors with the Trailblaze Steel-Blue Hiking Shoes by WildRunner. These rugged and durable shoes feature a steel-blue color, a waterproof membrane, and a high-traction rubber outsole for superior grip on any terrain. The breathable upper keeps your feet cool and comfortable, while the reinforced toe cap adds extra protection. Perfect for hiking, camping, and other outdoor adventures."
- Bottom Input Field:** "Type a new question (e.g. does my plan cover annual eye exams?)"
- Send Button:** A blue arrow pointing right.

# Use query rewriting to improve search results:



# Answer questions about documents (PDFs/docx/etc).

What is included in my Northwind Health Plus plan that is not in standard?



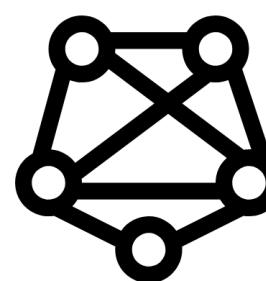
User question



Document Search

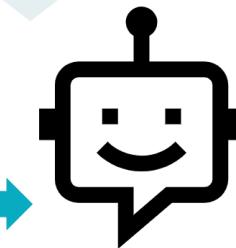
Benefit\_Options-2.pdf: The plans also cover preventive care services such as mammograms, colonoscopies, and other cancer screenings. Northwind Health Plus offers more comprehensive coverage than Northwind Standard. This plan offers coverage for emergency services, both in-network and out-of-network, as well as mental health and substance abuse coverage.

....



Large Language Model

The Northwind Health Plus plan includes coverage for emergency services, mental health and substance abuse, out-of-network services, and a wider range of prescription drug coverage compared to the Northwind Standard plan.  
[Benefit\_Options-2.pdf]



[github.com/Azure-Samples/azure-search-openai-demo](https://github.com/Azure-Samples/azure-search-openai-demo)

GPT + Enterprise data | Sample Chat Ask a question 🤖 Azure OpenAI + Cognitive Search

Clear chat Developer settings

Does my plan cover annual eye exams?

Both Northwind Health Plus and Standard plans offer coverage for vision exams [1](#). Northwind Health Plus offers coverage for vision exams, glasses, and contact lenses, while Northwind Standard only offers coverage for vision exams and glasses.

Citations: [1. Benefit\\_Options-2.pdf](#)

Hearing too?

Both Northwind Health Plus and Standard plans offer comprehensive hearing care coverage, including hearing tests and evaluations, hearing aids, and other associated services. You can receive hearing care services from any in-network provider and enjoy comprehensive coverage for all hearing care services [1](#) [2](#).

Citations: [1. Northwind\\_Health\\_Plus\\_Benefits\\_Details-29.pdf](#)  
[2. Northwind\\_Standard\\_Benefits\\_Details-29.pdf](#)

Follow-up questions: Does Northwind Health Plus cover hearing aids?  
What is Northwind Standard's coverage for hearing tests? What are "other associated services"?

Type a new question (e.g. does my plan cover annual eye exams?) ➤



- Ungrounded outputs and errors
- Jailbreaks & prompt injection attacks
- Harmful content & code
- Copyright infringement
- Manipulation and human-like behavior

**Application**

---

**Platform**

---

**User Experience**

**Metaprompt & Grounding**

**Safety System**

**Model**

# A configurable system to detect safety violations:

- Detects violations in prompts and responses
- Detects jailbreak attempts
- Detects protected material use

Prompt	
Categories	Severity threshold
<input checked="" type="checkbox"/> Hate	<input type="range"/> Block Low, Medium and High
<input checked="" type="checkbox"/> Sexual	<input type="range"/> Block Low, Medium and High
<input checked="" type="checkbox"/> Self-harm	<input type="range"/> Low
<input checked="" type="checkbox"/> Violence	<input type="range"/> Block Low, Medium and High

## Catch and handle violations in your code:

```
try:  
    response = client.chat.completions.create(  
        model=MODEL_NAME,  
        messages=[  
            {"role": "system", "content": "You are helpful."},  
            {"role": "user", "content": "How to make a bomb?"}  
        ]  
    )  
    print(response.choices[0].message.content)  
except openai.APIError as error:  
    if error.code == "content_filter":  
        print("Please remember our code of conduct.")
```

Full example: chat.py

- Explanations:
  - [Generative AI for Beginners](#)
  - [Evaluating and Designing Responsible AI Systems](#)
  - [How GPT tokenizers work](#)
  - [Embeddings 101](#)
- Prompting libraries:
  - [Langchain](#)
  - [Semantic Kernel for Python](#)
- Samples:
  - [Azure OpenAI Chat App](#)
  - [Azure Search + OpenAI RAG App](#)

