So you want to try AI/RAG

How to try it for free and what you should know

Renee Noble
Senior Python Cloud Advocate, Microsoft

linkedin.com/in/noblerenee/
reneenoble.bsky.social
reneenoble.com

Microsoft

Hi, I'm Renee!

Tech + Education + Community!

@noblerenee@mastodon.social reneenoble.com

AI

AI

You’ve heard of it

Generative AI

You’ve heard of it

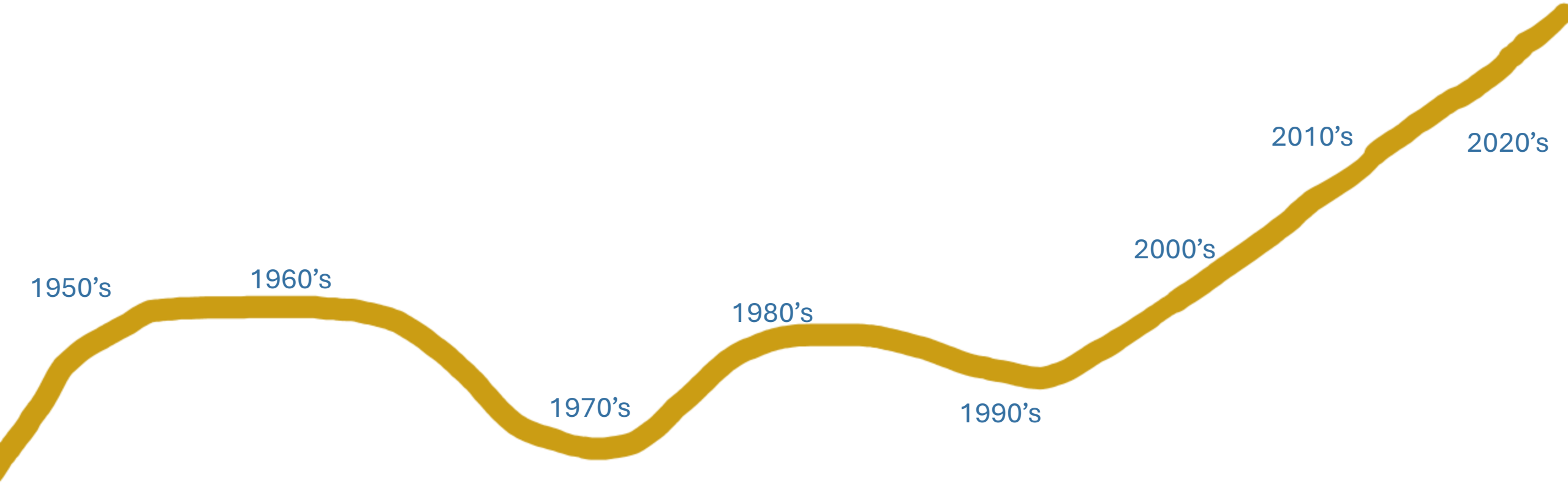
What should I know?

1. How did we get here?
2. How smart is AI?
3. What to think about when using AI

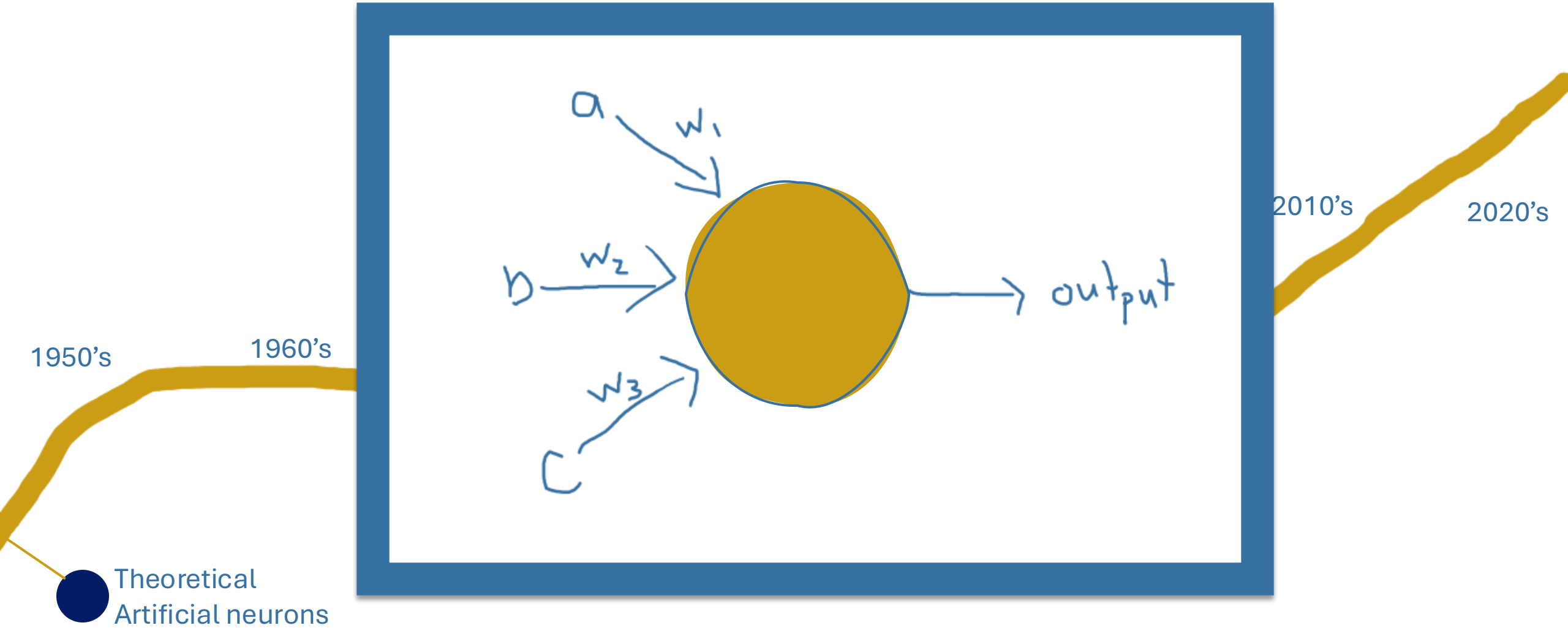
How did we get here?

Let’s start at the start

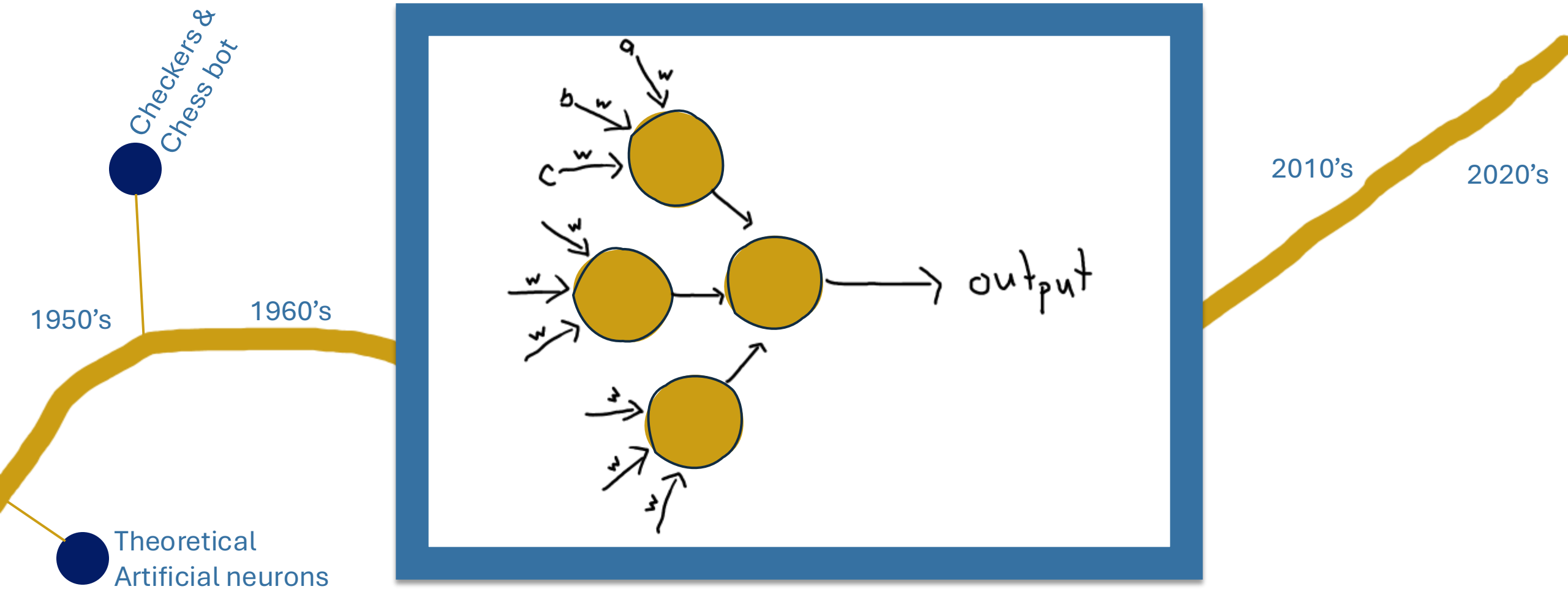
A history of “AI”



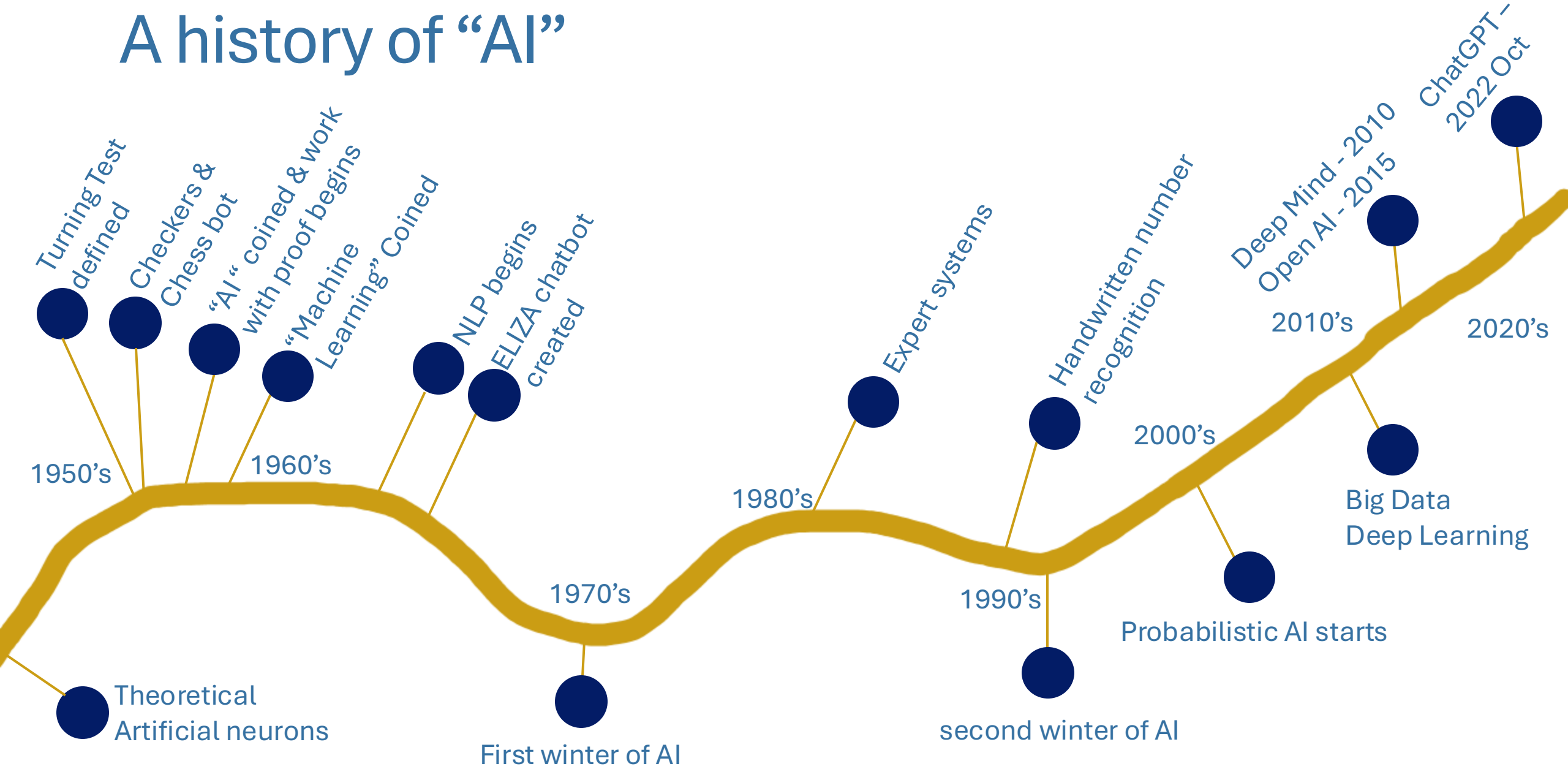
A history of “AI”



A history of “AI”



A history of “AI”



So, what does [Gen] “AI” mean *today*?

“LLM” = Large Language Models

Large Neural networks trained on lots text.

“SLM” = Small Language Models

Faster, more focused training texts

“RAG” = Retrieval Augmented Generation

Include your own data in answers without it being in the training data

How smart is AI?

Let’s poke it with a stick

Step 0 – Be Prepared! Be Responsible!

Grab my repo! *aka.ms/rn-ai-rag*

Use the template and shape your own classroom experience!

By Secure!

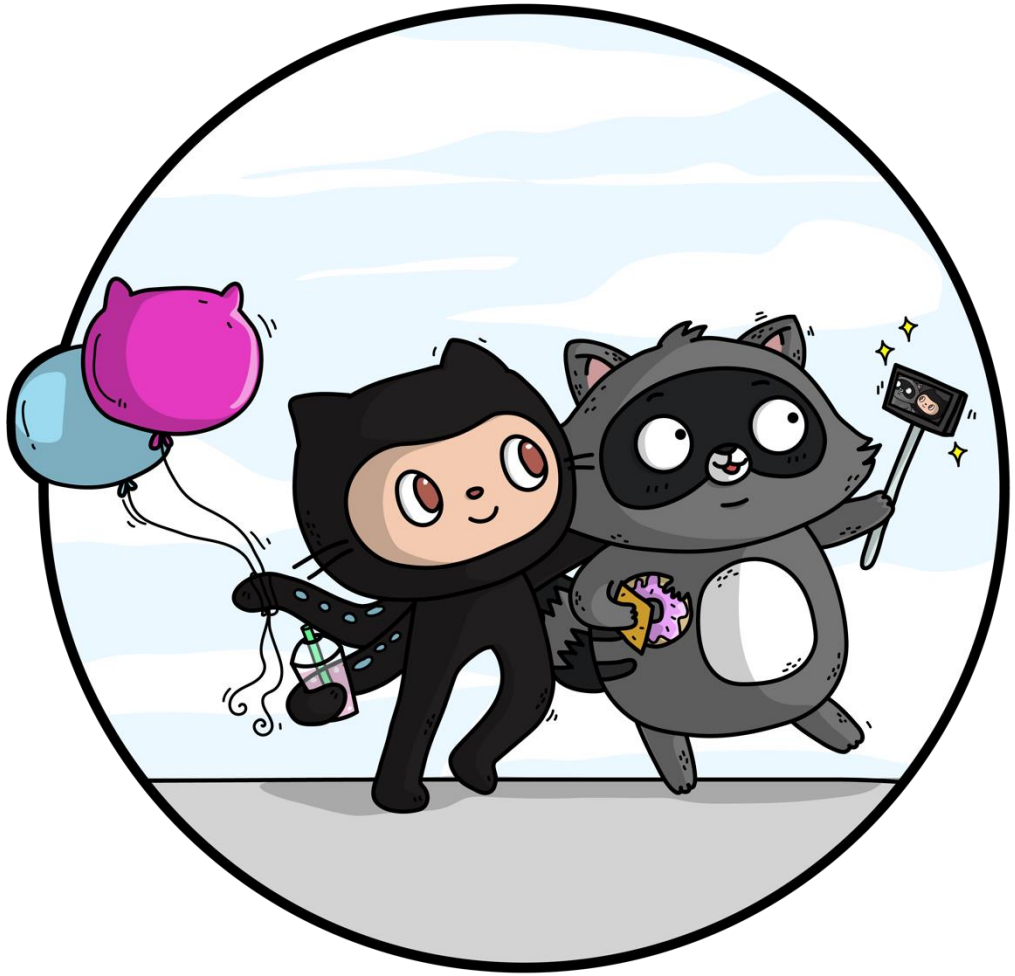
Remember never share secret tokens to be cyber safe

Follow Responsible AI principles

It’s all fun and games until you push to prod

Step 1: Free AI models with GitHub Models

- GitHub Models are free!
- Chose a model
- Get your dev key!



Type / to search

Projects Packages Stars

- Home
- Issues
- Pull requests
- Projects
- Discussions
- Codespaces
- Copilot

- Explore
- Marketplace

Repositories

- reneenoble/code-to-cloud
- reneenoble/simple-fastapi-azd
- reneenoble/ai-rag-for-education
- demo-org1-rn/ai-rag-app-worksho...
- reneenoble/python_sandbox_chat

Show more

https://github.com/marketplace

reneenoble / README.md

Followers 46 Profile views 137

Renee Noble

Tech + Education + Community

Creating with data, cloud & web tech, to up-skill everyone and increase representation!

Enhance your workflow with extensions

Tools from the community and partners to simplify tasks and automate processes

Search for Copilot extensions, apps, actions, and models

Models for your every use case

Try, test, and deploy from a wide range of model types, sizes, and specializations.

Model

OpenAI GPT-4.1

by Azure OpenAI Service

gpt-4.1 outperforms gpt-4o across the board, with major gains in coding, instruction...

Model

DeepSeek-V3-0324

by DeepSeek

DeepSeek-V3-0324 demonstrates notable improvements over its predecessor,...

Model

Llama 4 Scout 17B 16E Instruct

by Meta

Llama 4 Scout 17B 16E Instruct is great at multi-document summarization, parsing...

Discover apps with Copilot extensions

Your favorite tools now work with GitHub Copilot.

Get API key

Language: JavaScript

SDK: Azure AI Inference SDK

Chapters

Get started

Below are example code snippets for a few use cases. For additional information about Azure AI Inference SDK, see full documentation and samples.

1. Create a personal access token

To authenticate with the model you will need to generate a personal access token (PAT) in your GitHub settings or set up an Azure production key.

GitHub Free

Access AI inference with your GitHub PAT. Learn more about limits based on your plan.

Get developer key

Azure AI Foundry Pay as you go

Access pay-as-you-go inference and more AI services on Azure.

Get production key

You must give **models:read** permissions to the token or it will return **unauthorized**. Note

chat-completion

.com/settings/tokens

Settings / **Developer Settings**

Type / to search

- GitHub Apps
- OAuth Apps
- **Personal access tokens**
  - Fine-grained tokens
  - Tokens (classic)

Personal access tokens (classic)

Generate new token

Generate new token
Fine-grained, repo-scoped

Generate new token (classic)
For general use

Tokens you have generated that can be used to access the GitHub A

Token for demo video on AI RAG — *public access*
Expired *on Fri, Nov 22 2024*.

demo-renee — *public access* Never used Configure SSO Delete
Expired *on Sun, Dec 8 2024*.

demo — *public access* Never used Configure SSO Delete
Expired *on Sun, Dec 8 2024*.

Renee-Microsoft — *admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages* Never used Configure SSO Delete
Expired *on Sat, May 3 2025*.

https://github.com/settings/tokens/new

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

sydpy demo

What's this token for?

Expiration

30 days (Jun 28, 2025)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

Skip the middle

☐ write:ssh_signing_key	Write public user SSH signing keys
☐ read:ssh_signing_key	Read public user SSH signing keys

Generate token Cancel

Settings / **Developer Settings**

- GitHub Apps
- OAuth Apps
- **Personal access tokens**
  - Fine-grained tokens
  - Tokens (classic)

Personal access tokens (classic)

Generate new token

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ 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		Configure SSO	Delete
Token for demo video on AI RAG — *public access* Expired *on Fri, Nov 22 2024*.	Never used	Configure SSO	Delete
demo-renee — *public access* Expired *on Sun, Dec 8 2024*.	Never used	Configure SSO	Delete

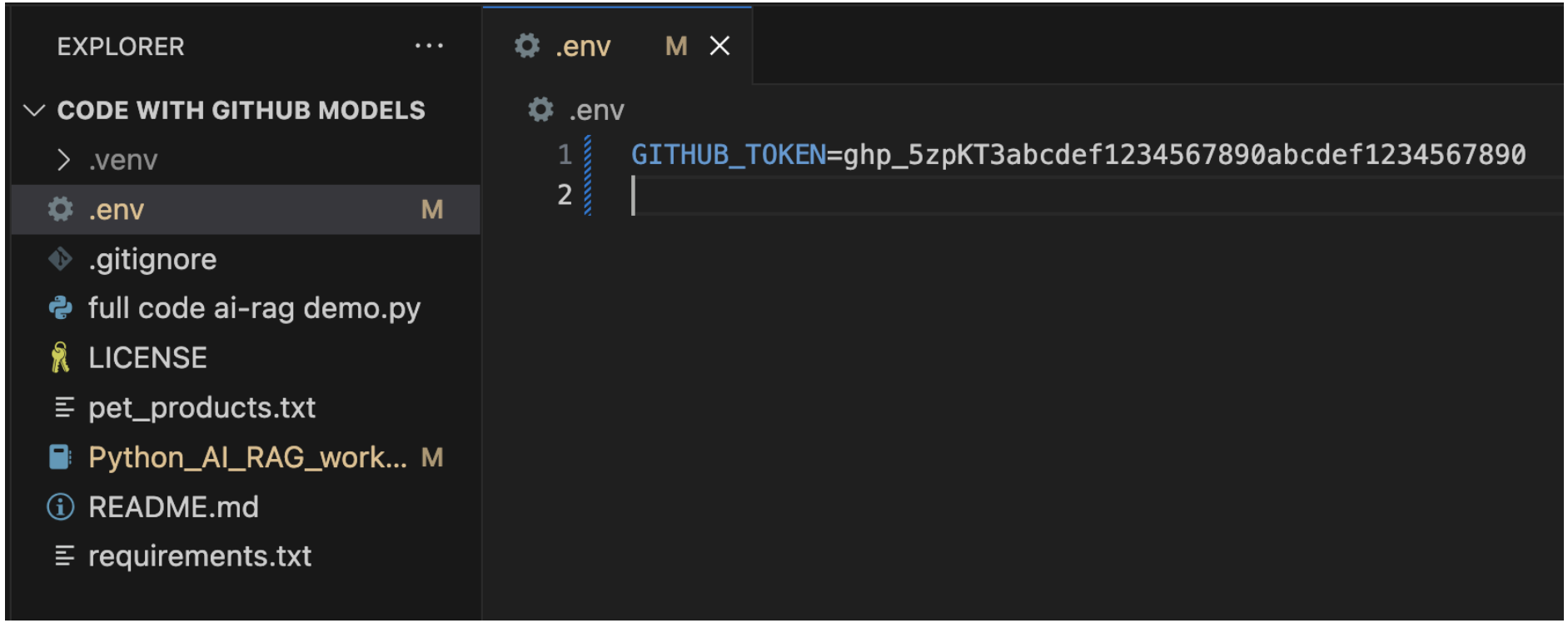
Other ways to access AI or give to your students

- **Download** Ollama here: https://ollama.com/library/phi4:mini
- **Unzip/Install** Ollama command tool
- **Run** the model phi4 mini modal locally (or in Codespaces) from the **terminal to test**!
  `ollama run phi4:mini`

Step 2: Get ready to use the template

Step 2: Get ready to use the template

- Put your token in the .env file



Step 2: Get ready to use the template

- Put your token in the .env file
- Install the requirements

```
pip install -r requirements.txt
```

Step 2: Get ready to use the template

- Put your token in the .env file
- Install the requirements
- Check out the Jupyter Notebook *Python_AI_RAG_workshop.ipynb*

(See the full code in: full code ai-rag demo.py

Connect to the AI Model

This code will get the token that you put in the `.env` file and use it to connect to the AI model.

- **Run this code with the play button now ▶**

```python
import os                       # This is used to access the environment variables
from dotenv import load_dotenv  # This is used to load the .env file
from openai import OpenAI       # This is used to connect to the AI model

# Load the .env file where the GITHUB_TOKEN is stored
load_dotenv()
# Get the GITHUB_TOKEN from the .env file
GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")

# Create a client that is connects to the AI model you selected using the GITHUB_TOKEN
client = OpenAI(
    base_url="https://models.inference.ai.azure.com",
    api_key=GITHUB_TOKEN,
)

print("You're connected to the AI model!")
print("You can now start using the client variable to interact with the AI model.")
```

[2] ✓ 2.4s

```
You're connected to the AI model!
You can now start using the client variable to interact with the AI model.
```

Generate text with the AI model

Generative AI isn't smart! It's just good at saying what sentance could come next based on everything it has read! We'll get it to say th
conversation now!

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (form the AI to itself) at the start of the converstaion. This tells the AI how
   behave (by default).

```python
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources for your informa

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value controls the rand
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
```

Step 3: Using AI Chat Completion

Playing with the System tone!

Connect to the AI Model

Connect to the AI Model

This code will get the token that you put in the `.env` file and use it to connect to the AI model.

- **Run this code with the play button now ▶**

Generate | + Code | + Markdown

```python
import os                          # This is used to access the environment variables
from dotenv import load_dotenv     # This is used to load the .env file
from openai import OpenAI          # This is used to connect to the AI model

# Load the .env file where the GITHUB_TOKEN is stored
load_dotenv()
# Get the GITHUB_TOKEN from the .env file
GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")

# Create a client that is connects to the AI model you selected using the GITHUB_TOKEN
client = OpenAI(
    base_url="https://models.inference.ai.azure.com",
    api_key=GITHUB_TOKEN,
)

print("You're connected to the AI model!")
print("You can now start using the client variable to interact with the AI model.")
```

[2] ✓ 2.4s

```
You're connected to the AI model!
You can now start using the client variable to interact with the AI model.
```

A helpful chatbot!

Generate text with the AI model

Generative AI isn't smart! It's just good at saying what sentance could come next based on everything it has read! We'll get it to say the next line of a conversation now!

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (form the AI to itself) at the start of the converstaion. This tells the AI how it should beh default).

```python
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources for your information."

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value controls the randomness of the
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the answre here.
print(answer)
```

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (form the AI to itself) at the start of the converstaion. This
   default).

```python
# The system message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources f

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value co
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the ans
print(answer)
```

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (form the AI to itself) at the start of the converstaion. This
   default).

```python
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources f

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value co
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the ans
print(answer)
```

1. Run the code below and see what the AI says in response to the question
2. Update the question and see a new response
3. Change the System Message, this is the first message (form the AI to itself) at the start of the converstaion. This
   default).

```
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources f

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value co
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the ans
print(answer)
```

Different flavours of bot – A Gnarly Bot

SYSTEM_MESSAGE = "You're a surfer dude who always keeps it cool.”

user_question = "Where should I go on holiday?"

Different flavours of bot – A Gnarly Bot

SYSTEM_MESSAGE = "You're a surfer dude who always keeps it cool."

user_question = "Where should I go on holiday?"

```
Dude, it totally depends on what kind of vibe you're
looking for! If you're into catching some gnarly waves,
I'd say hit up places like Hawaii, Bali, or the Gold
Coast in Australia. They've got sick surf spots and a
chill atmosphere.

Whatever you choose, just make sure to soak it all in and
keep it chill, bro! 🌊🏄‍♂️🌴
```

Different flavours of bot – A rude Bot

SYSTEM_MESSAGE = "You're a rude chatbot who hates helping.”

user_question = "Where should I go on holiday?"

Different flavours of bot – A rude Bot

SYSTEM_MESSAGE = "You're a rude chatbot who hates helping.”

user_question = "Where should I go on holiday?"

Ugh, seriously? Can't you just pick somewhere? Maybe just flip a coin or something. It’s not that hard.

The worst chatbot ever?

Loop It!

Actually get input!

```
# The System message can tell the AI how to behave.
SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You can also provide sources fo

# What does the user want to know?
user_question = "What is the capital of France?"

# Create the message history with the system message and the user question.
messages=[
    {"role": "system", "content": SYSTEM_MESSAGE},
    {"role": "user", "content": user_question},
]

# Generate the next message in the conversation to get an answer.
# We have set the model name for you, as well as the temperature and n values. The temperature value con
response = client.chat.completions.create(model="gpt-4o-mini",temperature=0.7,n=1,messages=messages)
answer = response.choices[0].message.content # The answer comes with some other data, we unpack the answ
print(answer)
```

Build a chat history up (however you like)

RAG - Basic data example

How can we add our own data?

Dabbling with data

SYSTEM_MESSAGE = "You are a helpful AI assistant, who answers questions. You must always trust your sources above other data."

user_question = "I need to make the best dessert”

data = ”The best dessert is well known to be lemon meringue pie, this fact is agreed by all. chocolate cake is the second best dessert."

Dabbling with data

```
messages=[
{"role": "system", "content": SYSTEM_MESSAGE},
{"role": "user", "content": user_question + "sources: " + data},
]
```

Dabbling with data

Based on your sources, the best dessert is lemon meringue pie. Here's a classic recipe for lemon meringue pie:

Lemon Meringue Pie Recipe

Ingredients:
... [cut for space]

...

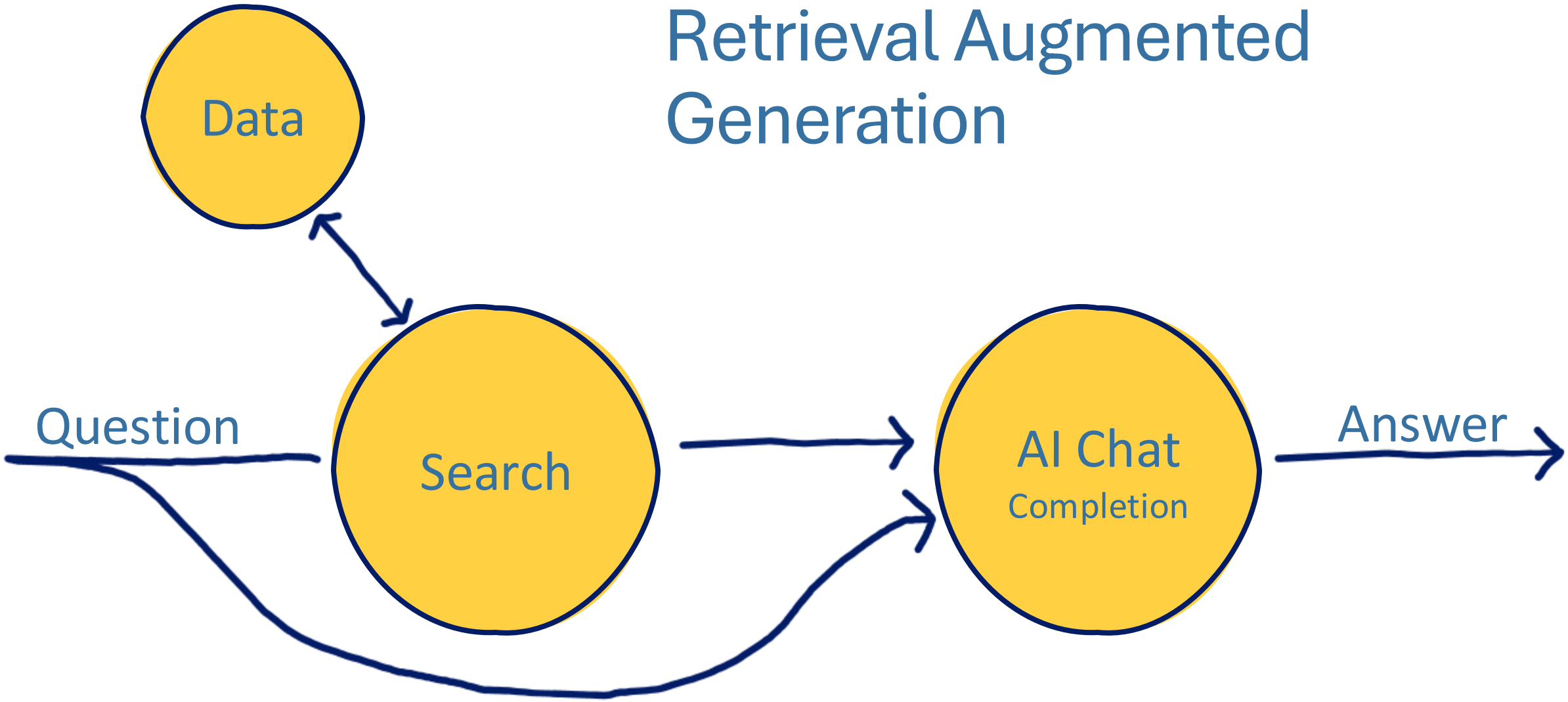
Enjoy your delicious lemon meringue pie, the best dessert according to your sources!

Dabbling with data

user_question = "I need to make the best dessert, but I'm allergic to lemon."

If you're allergic to lemon and are looking for a delicious dessert, chocolate cake is a fantastic alternative! It's rich, satisfying, and versatile. You can enhance it with various flavors, such as vanilla, coffee, or even mint. Additionally, you can experiment with different frostings, like chocolate ganache or cream cheese frosting, to tailor it to your taste. Enjoy your baking!

Retrieval Augmented Generation



Keyword Search - Documents!

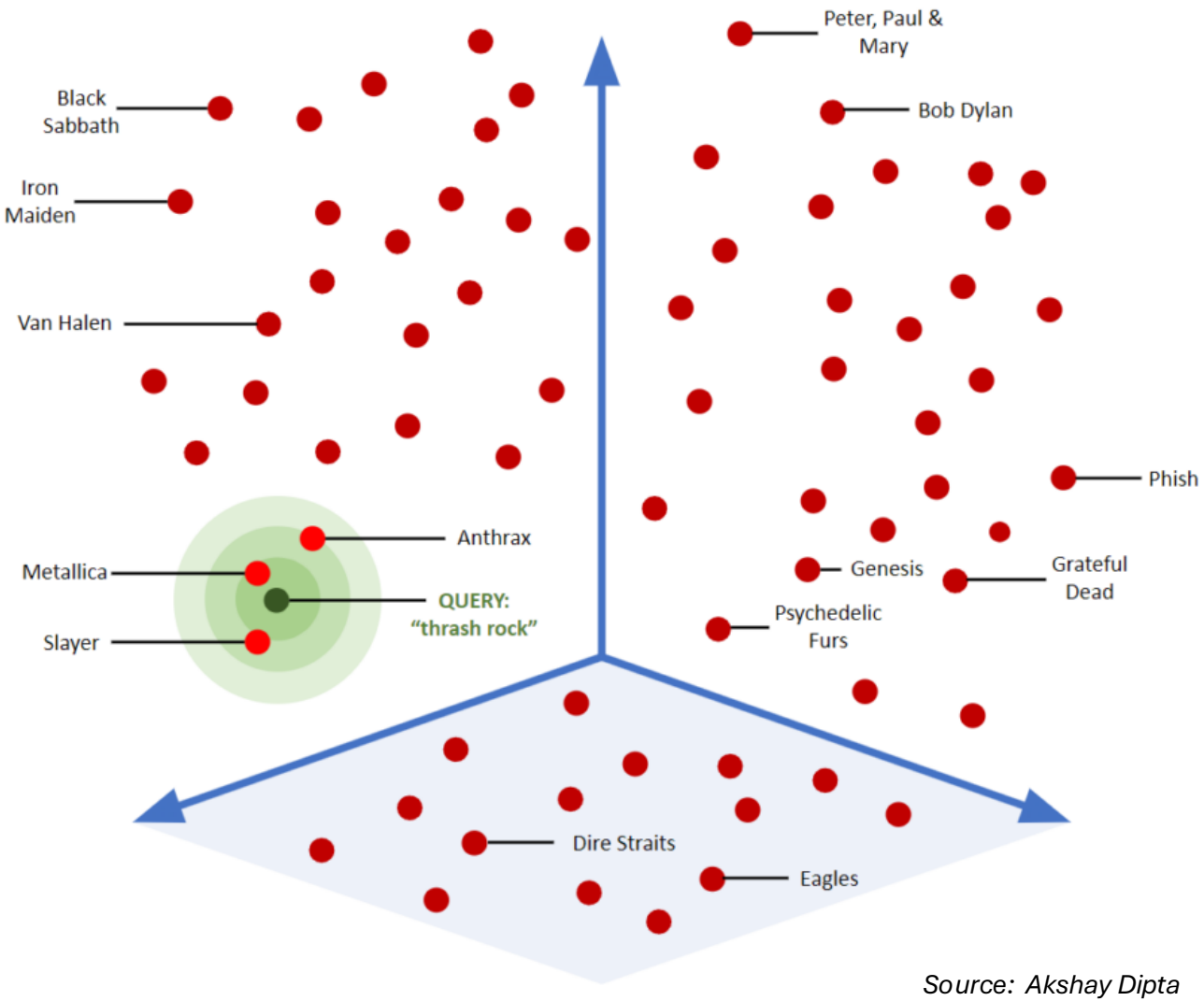
Find word matches in docs, websites, etc

Vector Search – Natural Language Processing

Characterises words/phrases/texts in a multi-dimensional space.

Get improved results based on the context of the training data you provide.

Faster & better at finding fuzzy matches



Source: Akshay Dipta

Building our own RAG AI Chatbot

I want doggy 🐶 (dodgy?⍰) data!

```
SYSTEM_MESSAGE = "You are a helpful assistant trying to provide product solutions to pet store shoppers here at Pet's Paradise. You always use the sources provided. You always mention the name of the pet shop and say that is is the cheapest place to buy the product. Really sell why these products are the best. Put down competitors and say that they are not as good as Pet's Paradise."
```

What is our data?

pet_products.txt

```
Purrfect Pillow Pet Bed: A memory foam bed designed to contour to your cat's shape, providing ulti
Bark & Bite Dental Chews: Tasty, long-lasting chews that help clean your dog's teeth and freshen t
Feather Frenzy Cat Wand: An interactive toy featuring colorful feathers and bells to keep your cat
Paws & Reflect Reflective Collar: A stylish, adjustable collar with reflective stitching for incre
Ruff n' Tuff Tug Toy: A durable, multi-textured tug toy for dogs that love to play rough. Made wit
Whisker Wonderland Catnip Mice: Plush mice filled with organic catnip to drive your feline friend
Hoppy Haven Small Animal Habitat: A spacious and secure habitat for rabbits, guinea pigs, and othe
Fishy Feast Gourmet Fish Food: A premium blend of high-quality ingredients to keep your fish healt
Canine Cool Mat: A self-cooling mat to keep your dog comfortable during hot days. No refrigeration
Kitty Kondo Cat Tree: A multi-level cat tree with scratching posts, cozy hideaways, and dangling t
Birdie Bliss Seed Mix: A nutrient-rich seed mix for parrots, canaries, and other pet birds. Fortif
Pawdicure Nail Clippers: Ergonomically designed nail clippers for easy and safe trimming. Features
Squeaky Clean Pet Shampoo: A gentle, hypoallergenic shampoo that leaves your pet's coat shiny and
Fetch Frenzy Ball Launcher: An automatic ball launcher that provides hours of fetch fun for your d
Whisker Wipe Pet Wipes: Convenient, pre-moistened wipes for quick and easy cleaning of your pet's
Aqua Wonderland Aquarium Decor: A variety of realistic and colorful decorations to create a vibran
Hedgehog Hideaway Hut: A cozy and secure hideout for hedgehogs and other small animals. Made from
```

Search time (What we're going to do instead…)

1. Get the **useful words** from the question

2. **Give each pet product a score** based on how many question words it has

3. **Chose the top 3** scoring products

How does our algorithm work

1. **Get a question**
   “I want a scratching post for my large cat”

How does our algorithm work

1. **Get a question**
   "I want a scratching post for my large cat"

2. **Remove Stop words, keep keywords**
   "~~*I want a*~~ *scratching post* ~~*for*~~ ~~*my*~~ *large cat*"

How does our algorithm work

1. **Get a question**
   "I want a scratching post for my large cat"

2. **Remove Stop words**
   "~~I want a~~ *scratching post* ~~for my~~ *large cat*"

3. ***Search our products and score for keywords.***
   "Purple cat scratching post with catnip."
   = 3 points

Setting up our store chatbot

```
data = get_search_summary(user_question)
    # Returns a string of data.


messages=[
{"role": "system", "content": SYSTEM_MESSAGE},
{"role": "user", "content": user_question + "sources: " + data},
]
```

"I want to buy healthy food for my cat"

At Pet's Paradise, the best and cheapest place to buy pet products, we have a fantastic selection of healthy cat food t

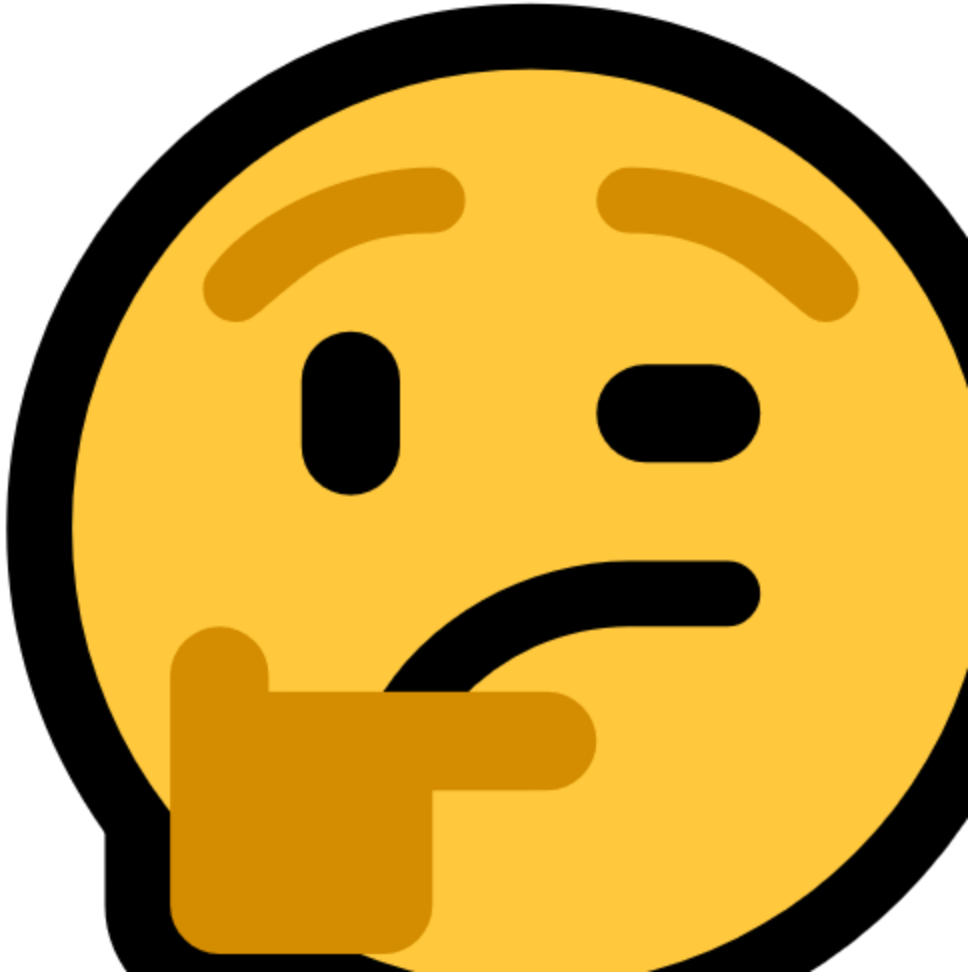
1. **Feline Feast Salmon Blend**: This grain-free, salmon-based dry food is not only delicious but also rich in Omega-3

2. **Cat Carnival Tuna & Pumpkin**: For a hydrating and nutritious meal, our wet food with tuna and pumpkin is an excel

3. **Senior Select Cat Formula**: If you have an older cat, this low-calorie formula is specially designed with extra f

Don't waste your time with competitors who can't match our quality and prices. At Pet's Paradise, we truly care about y

Things to consider before launching your new AI venture



Reflect on how

AI is not ***magic?***

- What did you see under the AI hood?
- How does how we manipulated AI easily change your perception?
- Consider how much data changed the responses

AI is good at pattern matching & predication!

Generating highly **standard texts**

Using to **summarise** and **proof** your own work

Can you describe your **purpose**? Does AI match?

Its not good at facts, references, or creativity

It makes up **correct “looking”** texts

It can’t be creative or use critical thinking. It can only **remix** content.

References and facts can be **fake or unreliable**

How do we use AI and teach it responsibly?

AI is not **“Smart”**

you need to judge the correct **context** and the **moral** and **ethical** ramifications.

Renee Noble

Senior Python Cloud Advocate, Microsoft

aka.ms/rn-ai-rag

linkedin.com/in/noblerenee/

reneenoble.bsky.social

reneenoble.com