Spring AI

edForce

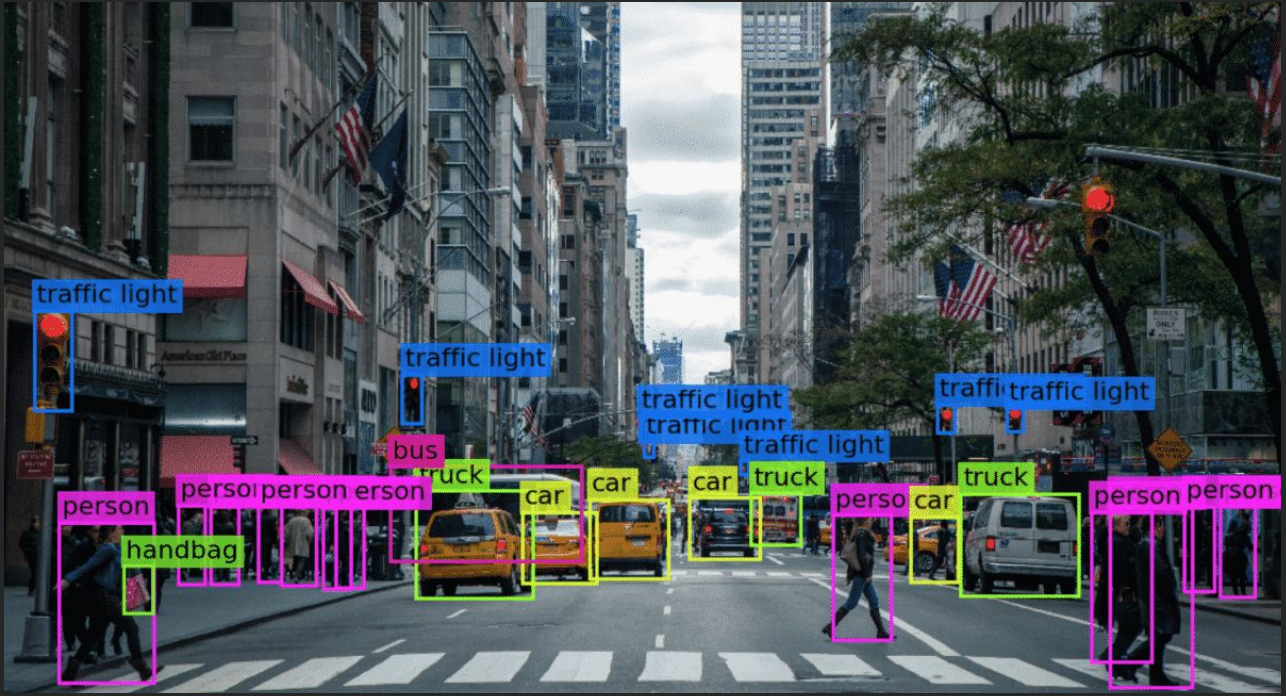
Who I am

- Michael Isvy 🇫🇷 🇸🇬

- Formerly part of SpringSource / Pivotal / VMware
  - Started teaching Spring in 2007
  - Started partnering with Ravi (Edforce CEO) in 2009

- VP of Engineering of cynapse.ai since 2022
  - Conventional AI, Generative AI, Computer Vision

Computer Vision (aka. Vision AI)



What we will talk about

- The AI Space
- Spring AI
- Retrieval Augmented Generation
- Vector Databases

Quiz

- There will be 4 quiz questions during this presentation
- Win some vouchers!

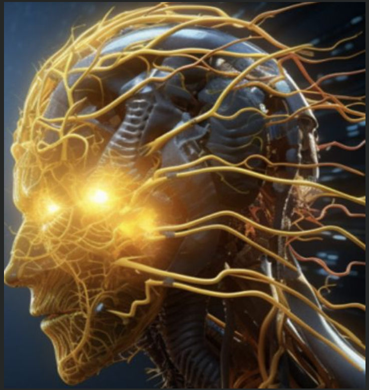
The AI space

edForce | WORKFORCE UPSKILLING ACCELERATED

Artificial Intelligence: which Way?

Conventional AI

Based on custom-trained models
Programming: requires AI engineers



Generative AI

LLM, ChatGPT, DALL-E, Gemini, Mistral, Ollama, Generative AI…
Programming: mostly API calls

Conventional AI example: Licence Plate Recognition

- Find a base model online

Typically on Github or HuggingFace

- Evaluate the model

Identify gaps (example: doesn't work with Singapore truck license plates)

- Prepare a fine-tuning dataset

- Spend 3-4 days training the model

Conventional AI

```python
model = MobileNetV2(weights='imagenet')


# Prepare a sample input image
img = image.load_img('path_to_image.jpg',
      target_size=(224, 224))
# …
# Run inference
predictions = model.predict(img_array)
# …
print(f"Predicted class: {predicted_class[1]}")
```

Python code

Load the model

Prediction

Conventional AI is typically done in Python or C++

Generative AI: usually an API call!

- Example: API call to ChatGPT

```bash
#!/bin/bash
API_KEY="sk-proj-7devtvnBIsYXVHJuHBQAT3BlbkFJNBB4uz8Iog5F2y"
curl https://api.openai.com/v1/chat/completions \
 -H "Content-Type: application/json" -H "Authorization: Bearer $API_KEY" \
 -d '{
 "model": "gpt-4o",
 "messages": [
   {"role": "user", "content": "Tell me a joke."}
 ]
}'
```

Linux/OSX

In GenAI, models are much more complex. But most of the time **you don't need to build them**

Quiz 1

- In the below example, there is something you should never do. What is it?

```bash
#!/bin/bash
API_KEY="sk-proj-7devtvnBIsYXVHJuHBQAT3BlbkFJNBB4uz8Iog5F2y"
curl https://api.openai.com/v1/chat/completions \
 -H "Content-Type: application/json" -H "Authorization: Bearer $API_KEY" \
 -d '{
 "model": "gpt-4o",
 "messages": [
   {"role": "user", "content": "Tell me a joke." }
 ]
}'
```

Linux/OSX

Generative AI

- General-Purpose models

- Use-cases
  - Chatbot
  - Image recognition (read invoices...)
  - Search a large number of documents
    - And ask questions from chatbot
  - ...

Quiz 2

- I work for a Payment company and we need to setup a system for Fraud detection. We will have a lot of custom rules in order to identify fraud patterns
- This is critical to our business
- Should I use Conventional AI or Generative AI for that?

The Generative AI landscape

In the Cloud

OpenAI
ChatGPT

Mistral AI
Mistral

Anthropic
Claude

Spin off from OpenAI

Google
Gemini

On premise / on your laptop

Ollama

Choose your local model

Llama3.1
Mixtral
LLava
tinyllama
…

Spring AI

What is Spring AI?

- AI for Java developers!
- Simplifies interactions with LLMs
  - change model by changing one line of configuration!
- Simplifies interactions with Vector databases

The Spring AI ecosystem

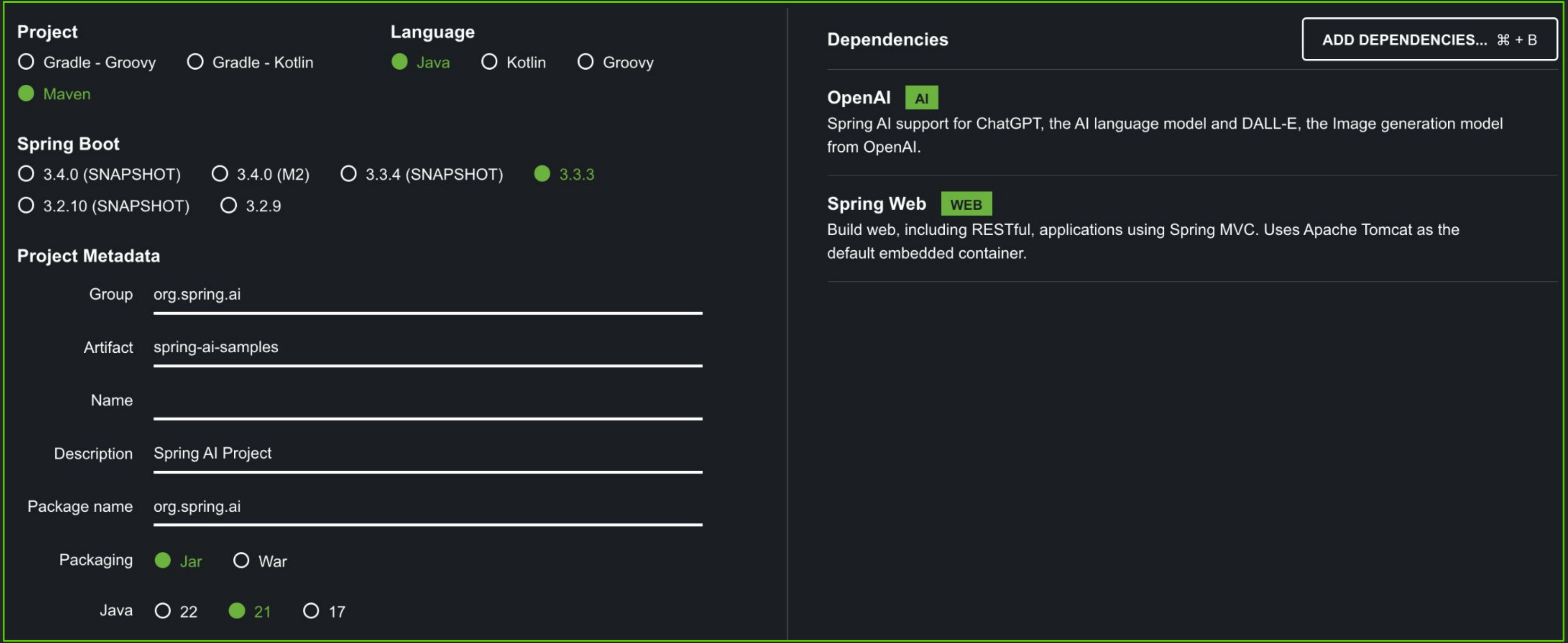
- Created in 2023 by Mark Pollack and Christian Tzolov
- Current version: Spring AI 1.0.0-M2
  - Not in final release version yet!
- Based on Spring and Spring Boot

Using Spring AI

- Create a Spring Boot Project
- Add the Spring AI dependencies
- Add your API key
- Use Spring AI to prompt queries to your model

Generating a Spring AI project

- Use start.spring.io



Project

○ Gradle - Groovy ○ Gradle - Kotlin [✓ Maven]

Language

[✓ Java] ○ Kotlin ○ Groovy

Spring Boot

○ 3.4.0 (SNAPSHOT) ○ 3.4.0 (M2) ○ 3.3.4 (SNAPSHOT) [✓ 3.3.3]
○ 3.2.10 (SNAPSHOT) ○ 3.2.9

Project Metadata

Group	org.spring.ai
Artifact	spring-ai-samples
Name	
Description	Spring AI Project
Package name	org.spring.ai
Packaging	[✓ Jar] ○ War
Java	○ 22 [✓ 21] ○ 17

Dependencies

OpenAI AI
Spring AI support for ChatGPT, the AI language model and DALL-E, the Image generation model from OpenAI.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Demo

Spring AI dependencies

```xml
<dependencies>
    <dependency>
        <groupId>org.springframework.ai</groupId>
        <artifactId> spring-ai-openai-spring-boot-starter </artifactId>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.ai</groupId>
            <artifactId> spring-ai-bom </artifactId>
            <version>1.0.0-M2</version>
            <type>pom</type>
            <scope>import</scope>
        </dependency>
    </dependencies>
</dependencyManagement>
```

Spring Boot starter (brings in all needed dependencies)

defines versions for all Spring AI dependencies (Bill Of Materials)

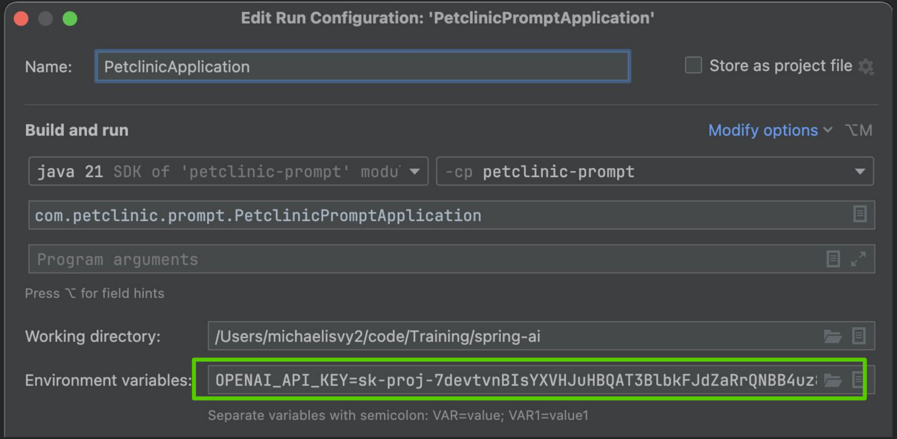
pom.xml

application.properties

- Best practice: store the API key is an env variable

```
spring.application.name=spring-ai-samples
spring.ai.openai.api-key=${OPENAI_API_KEY}
spring.ai.openai.chat.options.model=gpt-4o
```

application.properties



Edit Run Configuration: 'PetclinicPromptApplication'

Name: PetclinicApplication

☐ Store as project file

Build and run

```
java 21 SDK of 'petclinic-prompt' modu
-cp petclinic-prompt
com.petclinic.prompt.PetclinicPromptApplication
Program arguments
```

Press ⌥ for field hints

Working directory: /Users/michaelisvy2/code/Training/spring-ai

Environment variables: OPENAI_API_KEY=sk-proj-7devtvnBIsYXVHJuHBQAT3BlbkFJdZaRrQNBB4uz

Separate variables with semicolon: VAR=value; VAR1=value1

IntelliJ Community Edition

Making a call

- Use ChatClient's fluent API

```
@Service public class MusicService {
   private final ChatClient chatClient;

   public MusicService(ChatClient.Builder builder) {
     this.chatClient = builder.build();
   }

   public String findBestSongs() {
        return this.chatClient.prompt()
                 .user("which were the best songs in 1993?")
                 .call().content();
}  }
```

Generic (does not depend on the LLM implementation)

Adding a system prompt

- Give generic guidance to the prompt

```java
@Service
public class MusicService {
   private final ChatClient chatClient;
   public MusicService(ChatClient.Builder builder) { this.chatClient = builder.build(); }

   public String findBestSongs() {
       return this.chatClient.prompt()
                 .system(" You are a helpful assistant writing in English from the 1990s ")
                 .user("which were the best songs in 1993?")
                 .call().content();
   }
}
```

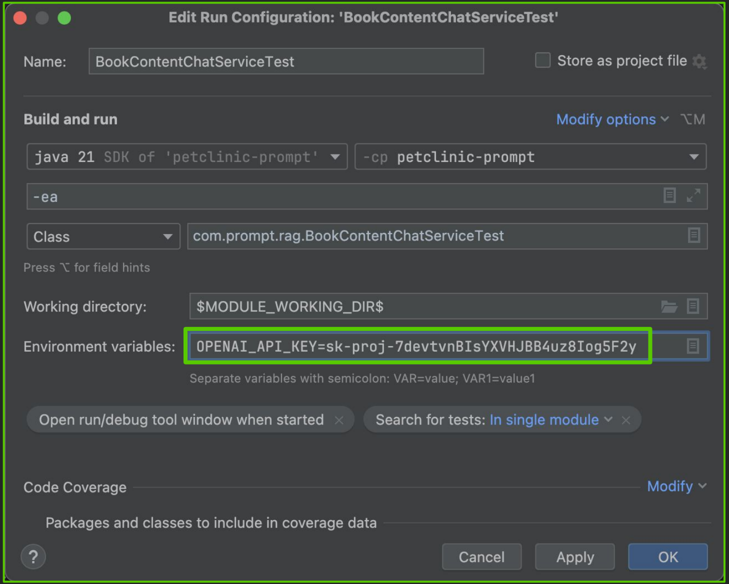
Calling from a JUnit test

- Run configuration should include the environment variable

```java
@SpringBootTest class MusicServiceTest {
   @Autowired
   private MusicService musicService;

   private static final Logger logger
            = LoggerFactory.getLogger(MusicService.class);

   @Test
   void shouldFindBestSongs() {
        String response = this.musicService.findBestSongs();
        logger.info(response);
}    }
```



Edit Run Configuration: 'BookContentChatServiceTest'

Name: BookContentChatServiceTest   ☐ Store as project file

Build and run   Modify options ⌥M

java 21 SDK of 'petclinic-prompt'   -cp petclinic-prompt

-ea

Class   com.prompt.rag.BookContentChatServiceTest

Press ⌥ for field hints

Working directory: $MODULE_WORKING_DIR$

Environment variables: OPENAI_API_KEY=sk-proj-7devtvnBIsYXVHJBB4uz8Iog5F2y

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started   Search for tests: In single module

Code Coverage   Modify

Packages and classes to include in coverage data

Demo

OpenAI vs Ollama

- OpenAI config

```xml
<dependency>
   <groupId>org.springframework.ai</groupId>
   <artifactId>
    spring-ai-openai-spring-boot-starter
   </artifactId>
</dependency>
```

pom.xml

```properties
spring.ai.openai.api-key=${OPENAI_API_KEY}
spring.ai.openai.chat.options.model=gpt-4o
```

application.properties

- Ollama config

```xml
<dependency>
   <groupId>org.springframework.ai</groupId>
   <artifactId>
    spring-ai-ollama-spring-boot-starter
   </artifactId>
</dependency>
```

pom.xml

```properties
spring.ai.ollama.chat.model=tinyllama
```

application.properties

Demo

Using a prompt with Parameters

```java
var response = this.movieService.recommendMovie("computers");
this.logger.info(response);
```

```java
public String recommendMovie(String topic) {
   return this.chatClient.prompt()
            .user( userSpec -> userSpec.text("Can you recommend a movie about about {topic}")
                            .param("topic", topic))
            .call()
            .content();
}
```

```
Certainly! One highly regarded film that delves into the world of computers is "The Imitation Game"
(2014). This biographical drama stars Benedict Cumberbatch as Alan Turing, a pioneering computer
scientist and mathematician.
```

Quiz 3

- In which version of Java was the “var” keyword introduced?

```
var response = this.movieService.recommendMovie("computers");
this.logger.info(response);
```

Mapping a prompt to an entity

```java
@Service
public class ChatService {
   private final ChatClient chatClient;

   public ChatService(ChatClient.Builder builder) { this.chatClient = builder.build(); }

   public ActorFilms generateResponse() {
        return this.chatClient.prompt()
            .user("Generate the 10 most popular movies starring Bruce Willis")
            .call()
            .entity(ActorFilms.class);
   }
}
```

```java
public record ActorFilms(String actor, List<String> movies) {}
```

works with Java records

Demo

JSON schema under the hood

```java
public ActorFilms generateResponse() {
    return this.chatClient.prompt()
        .user("Generate the 10 most popular movies
                                   starring Bruce Willis")
        .call()
        .entity(ActorFilms.class);
}
```

```java
public record ActorFilms(String actor,
                           List<String> movies) {}
```

Do not include any explanations, only provide an RFC8259 compliant JSON response …

{ \"$schema\" :
\"https://json-schema.org/draft/2020-12/schema\",
\"type\" : \"object\", \"properties\" : { \"actor\" : {
\"type\" : \"string\" }, \"movies\" : { \"type\" :
\"array\", \"items\" : { \"type\" : \"string\" } }
}

Demo

Quiz 4

- The below API shows a chain of method calls. There is a special name for that kind of API. How is it called?

- Bonus: can you name 3 Java frameworks or components which use the same technique?

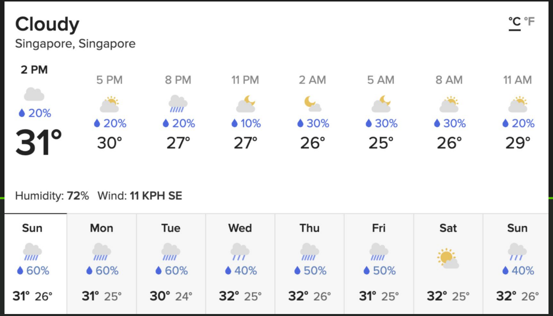
```java
public String findBestSongs() {
  return this.chatClient.prompt()
    .system(" You are a helpful assistant writing in English from the 1990s ")
    .user("which were the best songs in 1993?")
    .call().content();
}
```

Working with images

```
@Service class ImageService {
    @Value("classpath:images/singapore-weather.png")
    private Resource imageResourceWeather;
    // constructor
    public String analyseWeather() {
        return this.chatClient.prompt()
            .user(
              userSpec -> userSpec.text("what will be the weather like on Tuesday")
                                  .media(MimeTypeUtils.IMAGE_PNG, this.imageResourceWeather)
            )
            .call()
            .content();
    }
}
```

Import image file as a Resource

Optical Character Recognition



Cloudy
Singapore, Singapore
°C °F

2 PM	5 PM	8 PM	11 PM	2 AM	5 AM	8 AM	11 AM
20%	20%	20%	10%	30%	30%	30%	20%
31°	30°	27°	27°	26°	25°	26°	29°

Humidity: 72% Wind: 11 KPH SE

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
60%	60%	60%	40%	50%	50%		40%
31° 26°	31° 25°	30° 24°	32° 25°	32° 26°	31° 25°	32° 25°	32° 26°

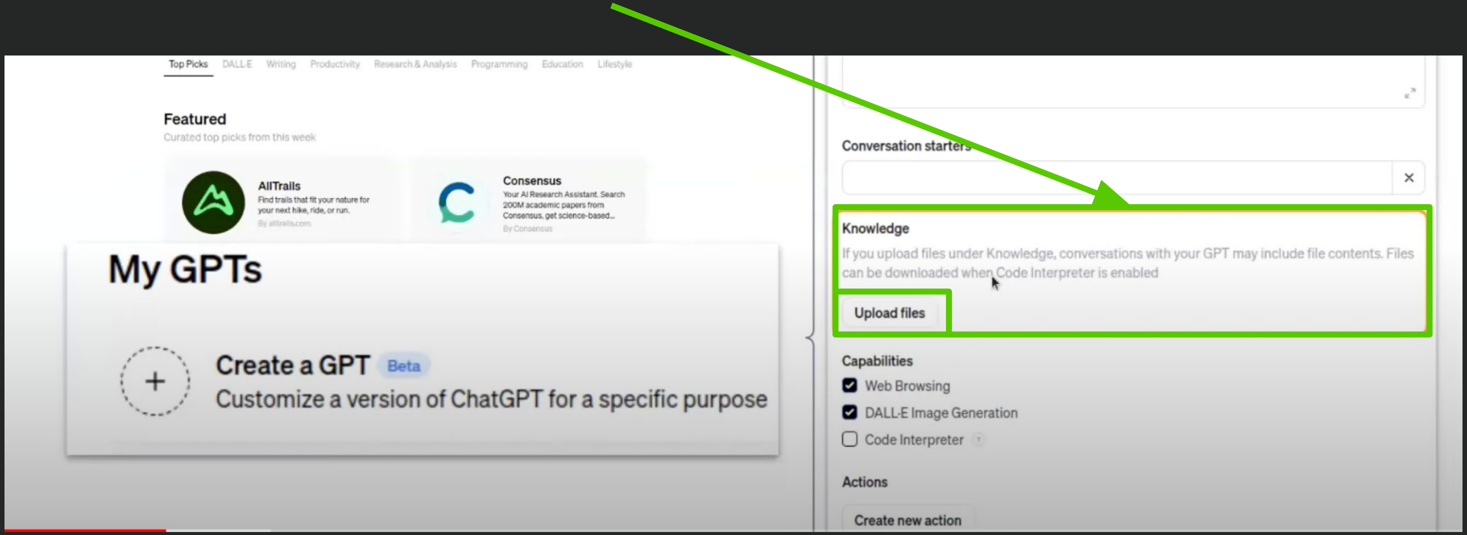
Demo

Retrieval Augmented Generation

edForce | WORKFORCE UPSKILLING ACCELERATED

Retrieval Augmented Generation

- Bring your own data to the prompt
- Give a lot of context to the prompt
  - Text content, excel, pdf, etc.
  - ChatGPT does it as well! (custom prompts)



Featured

Curated top picks from this week

AllTrails: Find trails that fit your nature for your next hike, ride, or run. By alltrails.com

Consensus: Your AI Research Assistant. Search 200M academic papers from Consensus, get science-based... By Consensus

My GPTs

Create a GPT: Customize a version of ChatGPT for a specific purpose

Conversation starters

Knowledge

If you upload files under Knowledge, conversations with your GPT may include file contents. Files can be downloaded when Code Interpreter is enabled

Capabilities

- [x] Web Browsing
- [x] DALL·E Image Generation
- [ ] Code Interpreter

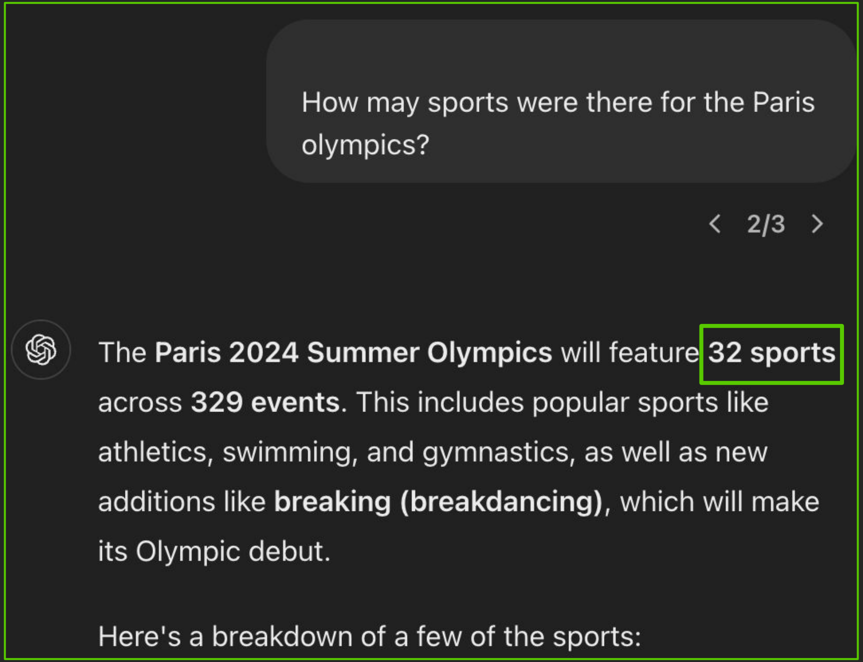
Actions

Why bringing your own data

- Models have only be trained on what is available on Internet

- Models are not real time
  - They all have a cutoff date
  - Example: ChatGPT-4o has been trained on data up to September 2023

Prompt data takes precedence over training data

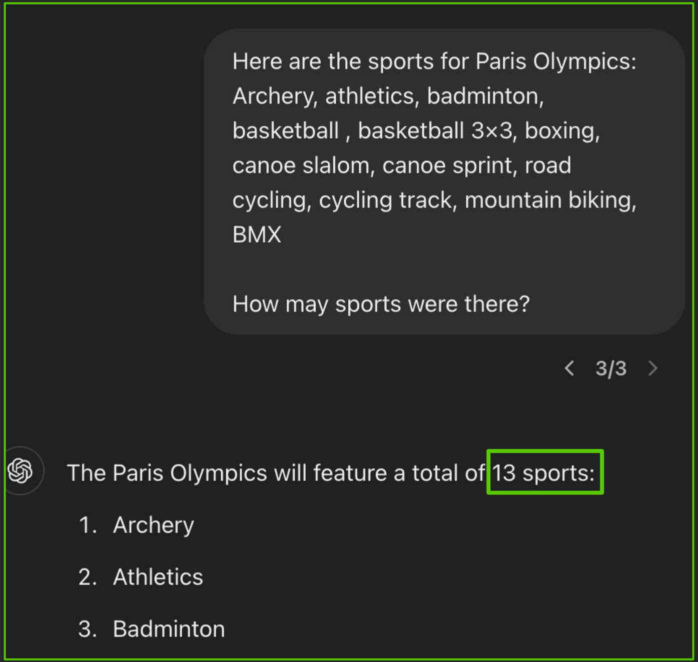


How may sports were there for the Paris olympics?

2/3

The **Paris 2024 Summer Olympics** will feature **32 sports** across **329 events**. This includes popular sports like athletics, swimming, and gymnastics, as well as new additions like **breaking (breakdancing)**, which will make its Olympic debut.

Here's a breakdown of a few of the sports:



Here are the sports for Paris Olympics: Archery, athletics, badminton, basketball , basketball 3×3, boxing, canoe slalom, canoe sprint, road cycling, cycling track, mountain biking, BMX

How may sports were there?

3/3

The Paris Olympics will feature a total of 13 sports:

1. Archery
2. Athletics
3. Badminton

Step 1 - Loading an st file into a Service class

```java
@Service
public class OlympicsService {

  @Value("classpath:/olympics/context.st")
  private Resource queryTemplate;
}
```

Structured text file

org.springframework.core.io.Resource

```
Use the following pieces of context to answer the question at the end.
{context}
Question: {question}
```

context.st

Step 2 - Using the prompt

```
@Value("classpath:/olympics/context.st")
private Resource queryTemplate;

public String findOlympicSports() throws IOException {
    return this.chatClient.prompt()
        .user( userSpec -> userSpec.text(this.queryTemplate)
            .param( "context" , "Archery, athletics, badminton, basketball , boxing")
            .param( "question" ,"How many sports are being included in the 2024 Summer Olympics?")
        )
        .call().content();
}
```

Structured text file

```
Use the following pieces of context to answer the question at the end.
Archery, athletics, badminton, basketball , boxing
Question: How many sports are being included in the 2024 Summer Olympics?
```

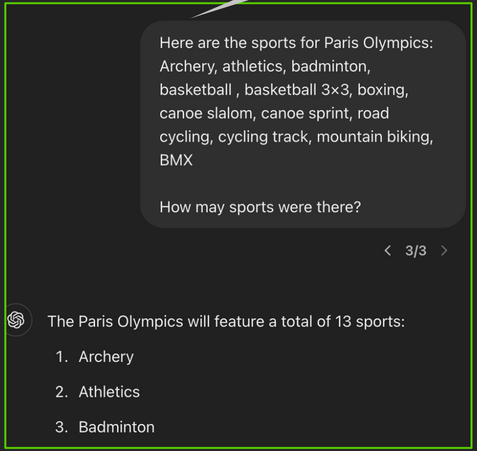
Demo

Vector databases

Quiz 5

- Using GPT-4's prompt context window, up to how many Harry Potter books can I fit at most? (talking about the last one that had over 700 pages)
  - 10% of a book
  - 50% of a book
  - 3 books

Context window

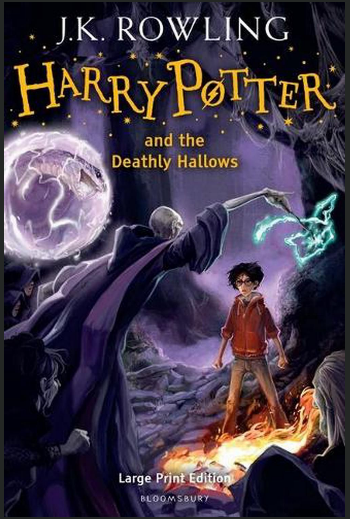


Here are the sports for Paris Olympics: Archery, athletics, badminton, basketball , basketball 3×3, boxing, canoe slalom, canoe sprint, road cycling, cycling track, mountain biking, BMX

How may sports were there?

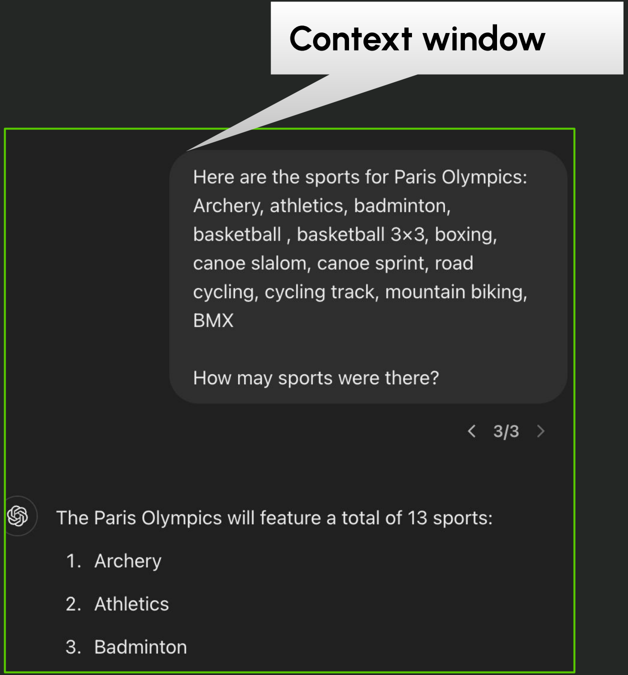
The Paris Olympics will feature a total of 13 sports:

1. Archery
2. Athletics
3. Badminton



Going beyond the prompt

- How to do when:
  - Your data is too big to fit into the prompt context window?
  - You're spending too much because of the prompt context



Here are the sports for Paris Olympics: Archery, athletics, badminton, basketball , basketball 3×3, boxing, canoe slalom, canoe sprint, road cycling, cycling track, mountain biking, BMX

How may sports were there?

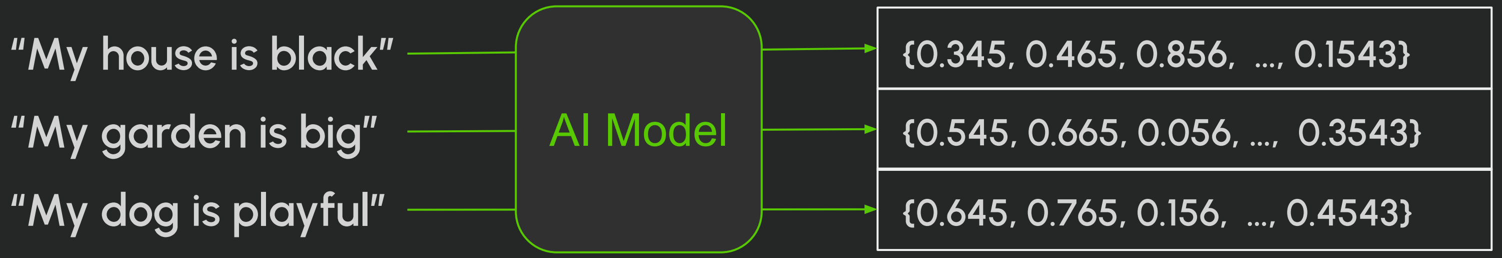
3/3

The Paris Olympics will feature a total of 13 sports:

1. Archery
2. Athletics
3. Badminton

Solution: Vector databases

- Split your data into chunks, and encode each chunk into numbers that the ML model can understand



Each Vector is an array of 1,536 numbers

Definition of a Vector

- A Vector is just a type of data
    - Typically an array of 1,536 decimal numbers
    - Value between -1 and 1

```
CREATE TABLE paragraph (
      id SERIAL PRIMARY KEY,
      paragraph_text TEXT,
      vector VECTOR(1536)
);
```

example with pgvector

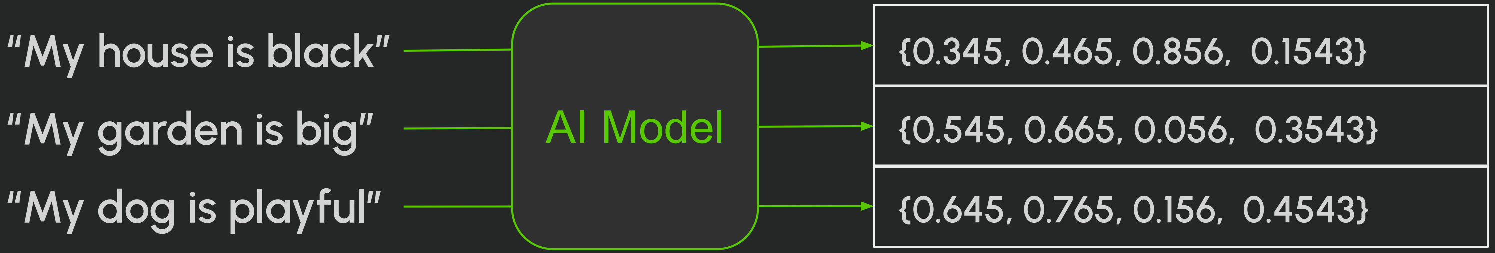
{-0.345, 0.465, 0.856, ..., 0.1543}

Each Vector is an array of 1,536 numbers

"Vector" and "Embedding" are similar concepts. For simplicity, we use the word "Vector" whenever possible in this course

How are Vectors created?

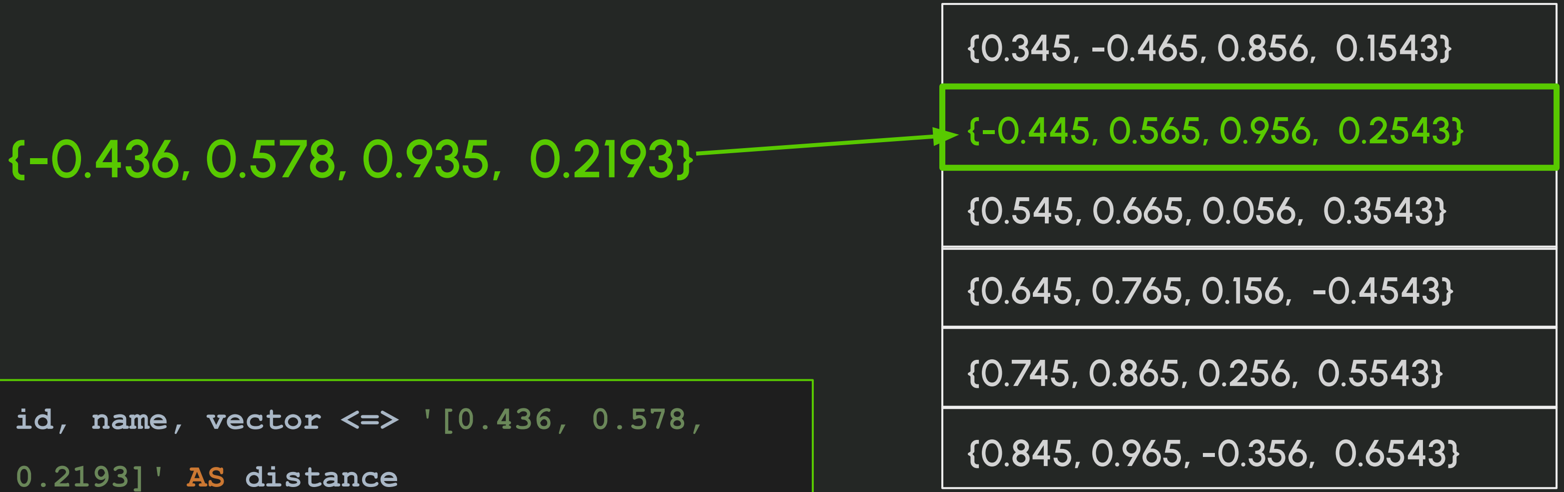
- Vectors typically represent the output generated by Machine Learning models



The above example is simplified and uses random numbers
It is based on ***text AI models****. Vectors may also be used with* ***Computer Vision AI models*** *or audio-based AI models*

How to search vectors: Similarity Search

- Selects the closest Vector(s)



```sql
SELECT id, name, vector <=> '[0.436, 0.578,
0.935, 0.2193]' AS distance
FROM items ORDER BY distance LIMIT 10;
```

sample SQL query with pgvector

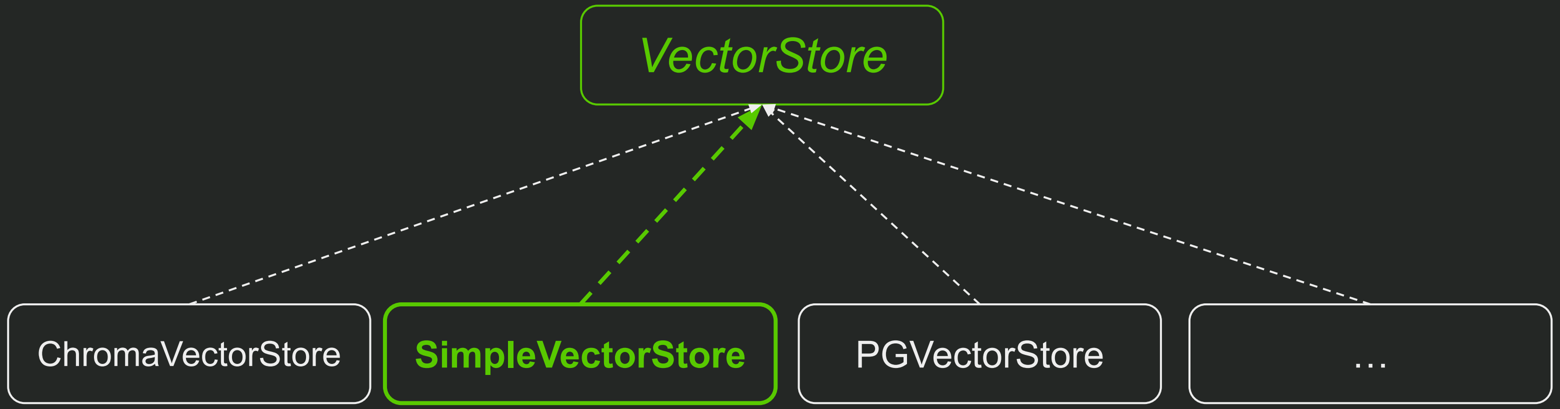
Vector database providers

- Most SQL and NoSQL databases are working on their Vector support
  - PostgreSQL (pgvector), MongoDB, ElasticSearch, Cassandra, …

- Some databases are specialised Vector databases
  - Chroma, Milvus, …

SimpleVectorStore

- Spring AI comes with a file-system based VectorStore implementation
    - To be used for Educational purpose only



SimpleVectorStore example with OpenAI

```
@Configuration
class VectorStoreConfiguration {
  @Bean
  SimpleVectorStore simpleVectorStore(EmbeddingModel embeddingModel) throws IOException {
    var simpleVectorStore = new SimpleVectorStore(embeddingModel);
    //...
    return simpleVectorStore;
} }
```

OpenAIEmbeddingModel is injected

```
{ "aec18bbc-21dc-4763-b93e-2f2ee49f9024" : {
   "embedding" : [ -0.0671, -0.0342, -0.0103, …],
   "content" : "He raised the guitar, and Henri …",
   "id" : "aec18bbc-21dc-4763-b93e-2f2ee49f9024",
   "metadata" : {
     "source" : "crime-in-paris.txt"
 }   }
```

sample vector.json file

Example: encoding a text into a Vector database

- Step 3: Similarity Search

```java
public String answerQuestion(String question) {
    return chatClient.prompt()
                .user(question)
                .call()
                .content();
}
```

1. Calls OpenAI Model in order to encode question
2. Compares question against all vectors inside database
3. Returns list of closest vectors
4. Sends Vector to ChatGPT together with question

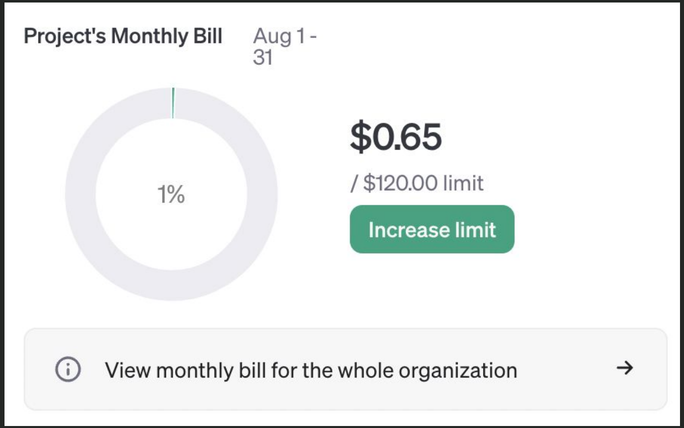
Demo

Conclusion

WORKFORCE
UPSKILLING
ACCELERATED

Cost of 1 month of experimenting with Spring AI

- OpenAI
  - $0.65


Project's Monthly Bill Aug 1 - 31

$0.65

/ $120.00 limit

View monthly bill for the whole organization

- Ollama
  - $0.00

Is Conventional AI dead?

- Absolutely not!

- Conventional AI is still the best when:
  - Working with a private or confidential dataset
  - There is high accuracy requirement requiring model fine-tuning

- Generative AI shines for:
  - Text generation
    Quick embedding of AI features in your application

As of now, ***Spring AI focuses on Generative AI***

My favorite Spring AI Resources online

- https://www.youtube.com/@DanVega (Dan Vega)
- https://www.youtube.com/@springinaction (Craig Walls)
- Videos from Spring I/O Barcelona conferences

- My demos: https://github.com/michaelisvy/demo-spring-ai

Questions and Answers

edForce | WORKFORCE UPSKILLING ACCELERATED