Building Applications Using Generative AI

Ram N Sangwan

- Role of Developers as Consumers of Generative AI APIs.
- Building Applications from Generative AI Outputs.
- Session & Chat History Management Best Practices
- Framework for Output Validation & Continuous Improvement of Prompts
- Deployment Options and Best Practices.

Role of Developers As Consumers of Generative AI APIs.

Generative AI Developer - Responsibilities

Model Selection and Architecture

- The quality of training data directly impacts the performance of a generative model.
- Curate and pre-process datasets to ensure they are diverse, representative, and free from biases that could affect the model's output.

Model Training and Fine-Tuning

- GenAI Developers manage the training phase, optimizing hyperparameters and adjusting the model's architecture to achieve the desired results.
- Fine-tuning is an iterative process that refines the model's output.

Evaluation and Validation

- GenAI Developers employ various metrics to evaluate the model's performance.
- They also conduct human evaluations to ensure that the generated content aligns with the intended purpose.

Generative AI Developer - Responsibilities

Deployment and Integration

- Once the model meets the quality standards, it's time to deploy it into applications.
- You may develop APIs, design user interfaces, or integrate the model into existing systems to make it accessible and user-friendly.

Enhanced Personalization

- GenAI Developers will focus on enhancing the personalization capabilities of AI models.
- This means AI-generated content will become more tailored to individual preferences, whether it's in marketing, content creation, or virtual assistants.

Ethical Considerations

- *With great power comes great responsibility.*
- You will play a pivotal role in addressing ethical concerns related to AI, such as bias, privacy, and the responsible use of AI-generated content.
- You need to ensure that AI models are trained and deployed in an ethical and transparent manner.

Generative AI Developer - Responsibilities

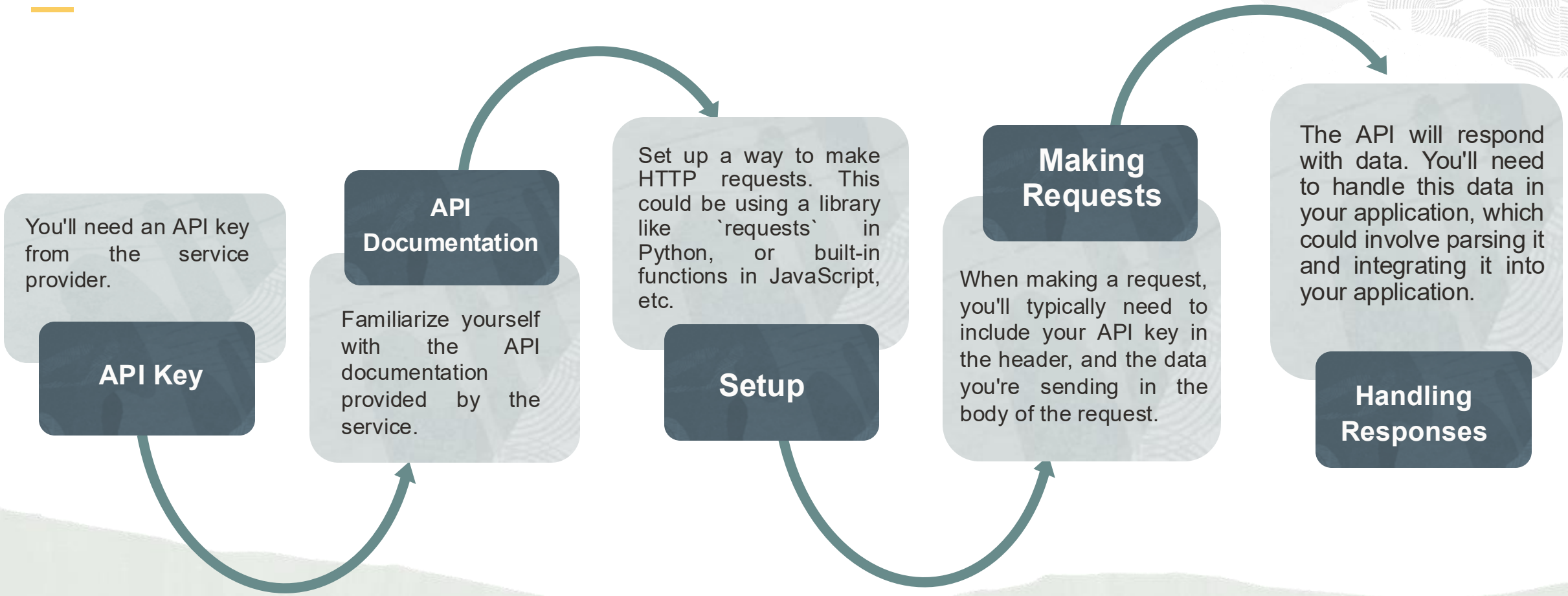
Cross-Domain Expertise

- The versatility of LLMs means that GenAI Developers will need to develop cross-domain expertise.
- They may work on projects ranging from healthcare and finance to entertainment and education, adapting their skills to various industries and applications.

API integration:

Connect and Make Requests to GenAI services.

Connect and Make Requests to Generative AI services.



Connect and Make Requests to GenAI Services.

A basic example in Python using the `requests` library:

Examples:

```
import requests
import json

url = "https://api.your-ai-service.com/generate"

headers = {
    "Authorization": "Bearer YOUR_API_KEY",
    "Content-Type": "application/json"
}
data = {
    "input": "Your input text here"
}
response = requests.post(url, headers=headers, data=json.dumps(data))
generated_text = response.json()["output"]

print(generated_text)
```

Define the endpoint URL

Define the headers

Define the data you're sending (this will depend on the API)

Make the POST request

Parse the response

The structure of `data` and the response handling will depend on the specific API you're using.

Building Applications that Leverage GenAI outputs

Building Applications that Leverage GenAI Output

Here are some of the areas of implementation

Automated Custom Software Engineering

Generative AI is leading the way, start-ups like GitHub's CoPilot are streamlining coding processes.

Content Generation with Management

Autonomous content generation is a powerful tool for any business, allowing them to create high-quality content faster and more efficiently than ever before while augmenting human creativity.

Marketing and Customer Experience

Businesses can generate high-quality content quickly and efficiently, saving time and resources. This can include blog articles, ad captions, product descriptions, and more.

Building Applications that Leverage GenAI Output

Healthcare

Mini Protein Drug Discovery and Development

Cancer Diagnostics

Paige AI has developed generative models to assist with cancer diagnostics, creating more accurate algorithms and increasing the accuracy of diagnosis.

Diagnostically Challenging Tasks.

Day-to-day Medical Tasks

Antibody Therapeutics

Absci Corporation uses machine learning to predict antibodies' specificity, structure, and binding energy for faster and more efficient development of therapeutic antibodies.

Building Applications that Leverage GenAI Output

Product Design and Development

GenAI provide innovative solutions that are too complex for humans to create.

It can help automate data analysis and identify trends in customer behaviour and preferences to inform product design.

GenAI allows for virtual simulations of products to improve design accuracy, solve complex problems more efficiently, and speed up the research and development process.

How to Build a Generative AI solution?

Requires a deep understanding of both the technology and the specific problem it aims to solve.

Involves designing and training AI models to generate novel outputs based on input data, often optimizing a specific metric.

Key steps includes defining the problem, collecting and pre-processing data, selecting appropriate algorithms and models, training and fine-tuning the models, and deploying the solution in a real-world context.

How to Build a Generative AI solution?

Problem Defining and objective setting

Comprehend the problem to be addressed and the desired outputs.

Data Labelling and Refinement

The **data refinement:** cleaning the data to eliminate errors, normalizing it and augmenting the dataset to improve its richness and depth.
Data labelling: involves manually annotating data to facilitate more effective AI learning.

Model training and fine-tuning

Deployment and monitoring

Deployment is not only technical; it also involves ethics.
Principles of transparency, fairness, and accountability guide the release into the real world.

Before training, one needs data and lots of it. This process entails gathering vast datasets and ensuring their relevance and quality.
Data should be from diverse sources and stripped of any copyrighted content.

Data collection and management

Foundational models.
Starting with an existing model, leverage the inherent capabilities and further fine-tune it for specific tasks.

Choosing a foundational model

Model Evaluation: a pivotal step to ascertain the model's performance after training.
This process ensures the model achieves the desired results and is reliable in varied scenarios.

Model evaluation and refinement

Session & Chat History Management Best Practices

Session & Chat History Management Best Practices

These are crucial aspects of Generative AI, especially in applications like chatbots or virtual assistant

Here are some best practices:

- **Data Privacy and Security**: Always ensure that the data collected during sessions is stored securely and in compliance with relevant data protection regulations. Personal data should be anonymized or pseudonymized to protect user privacy.
- **Data Retention**: Implement a data retention policy that specifies how long session data is stored.
- **Context Management**: Maintain the context of the conversation in the session history.

Session & Chat History Management Best Practices

- **Error Handling**.
- **Feedback Loop**.
- **Clear Communication**: Inform users about how their data will be used and stored.
- **Scalability**: to handle increasing amounts of chat data as the user base grows.
- **Multi-Session Management**: For AI models that interact with the same users over multiple sessions, the system should be able to link these sessions together to provide a more personalized and consistent user experience.

Framework for Output Validation & Continuous Prompts Improvement

Output validation and continuous prompts improvement cycle

Output Validation

- It involves using various metrics to assess the performance of the AI model.
- Human evaluation is also crucial to ensure the generated content aligns with the intended purpose and is contextually appropriate.

Error Analysis

- This could involve examining instances where the model's output deviated from the expected result, or where the model's performance was subpar.

Model Improvement

- Based on the error analysis, the model is then improved.
- This could involve fine-tuning the model, adjusting hyperparameters, or even changing the model architecture.

Output validation and continuous prompts improvement cycle.

Prompt Engineering

- The aim is to guide the model to produce better outputs.

Retraining and Re-evaluation

- The improved model is then retrained, and the newly generated outputs are re-evaluated.

Deployment

- Once the model's performance meets the desired standards, it is deployed. However, the improvement cycle doesn't stop here.

This framework ensures that the Generative AI models are continuously improved and adapted to generate high-quality, relevant, and contextually appropriate content.

Deployment Options

Deployment Options for Generative AI

Cloud-Based Deployment

Many LLMs are deployed on cloud platforms like Oracle Cloud, AWS, Google Cloud, or Azure.

On-Premises Deploy

For sensitive applications where data privacy is paramount, LLMs can be deployed on-premises.

This requires substantial computational resources and expertise.

Edge Deployment

In some cases, LLMs can be deployed on edge devices (like smartphones or IoT).

This requires model optimization to reduce resource usage.

https://blogs.nvidia.com/blog/what-is-edge-ai/

Thank You