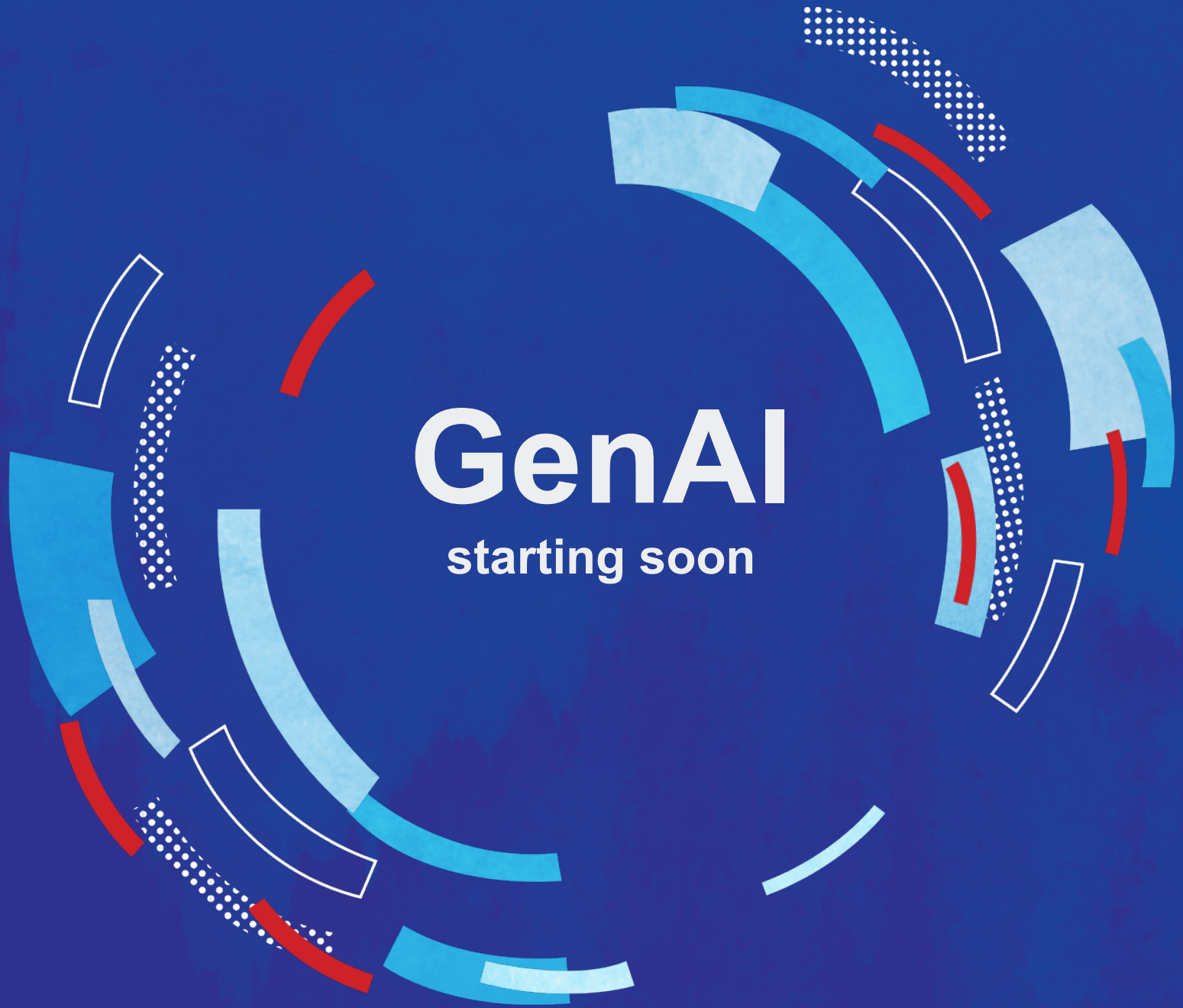
GenAI

starting soon



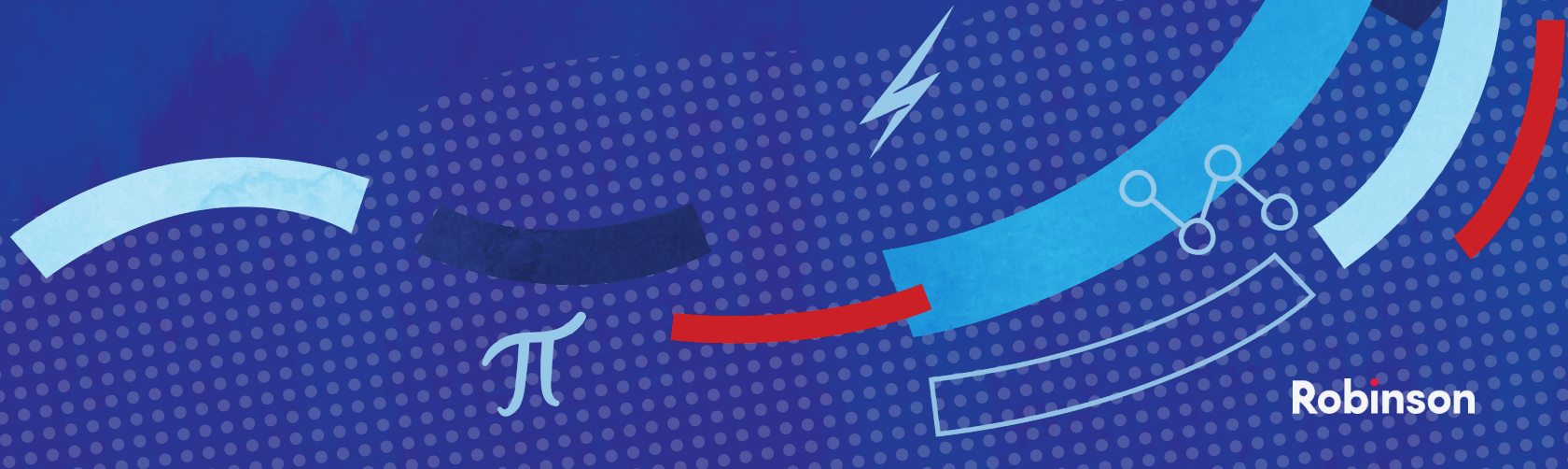
Georgia State University | J. Mack Robinson College of Business

Robinson

Generative AI Workshop

Péter Molnár <pmolnar@gsu.edu>

Fall, 2024



Georgia State University | J. Mack Robinson College of Business

Robinson

Péter Molnár is an associate professor at Georgia State University's J. Mack Robinson College of Business where he teaches artificial intelligence, focusing on both technical implementation and executive decision-making. His career bridges academia and industry, including work as a data scientist at Amazon Web Services on innovative machine learning applications

pmolnar@gsu.edu

https://www.linkedin.com/in/petermolnar/

Agenda/Activities

2:00	Introductions
2:05	Review Prompt Engineering and AI Agents
2:10	Overview of Activities
	Form Teams – Pick Activit[y|ies]
3:45	Groups share their outcomes
	Discussion/Questions
4:30	Adjourn

https://github.com/molnarai/https://github.com/molnarai/genai-exercises

Prompt Engineering

Image created with
Microsoft Copilot

Robinson

Prompt Engineering

Prompt engineering in large language models (LLMs) refers to the strategic crafting of input prompts to guide the model's generation of outputs.

This process is essential because LLMs, such as GPT-3 or BERT, are trained to predict the next word or sequence of words based on the input they receive. The quality and structure of the prompt can significantly influence the relevance, coherence, and accuracy of the model's response.

LLMs are dependent on prompts as they serve as the interface for human-model interaction, providing context and instruction that shape the model's behavior.

Without well-designed prompts, LLMs may generate outputs that are off-topic, factually incorrect, or fail to grasp the user's intent.

Effective prompt engineering can mitigate these issues by incorporating clear instructions, examples, or structured queries that align the model's responses with desired outcomes, making it a critical skill for leveraging the full potential of LLMs in various applications

Elements of a Prompt

Persona: Describe what role the LLM should take on. For example, use “You are an expert in astrophysics” if you want to ask a question about astrophysics.

Instruction: The task itself. Make sure this is as specific as possible. We do not want to leave much room for interpretation.

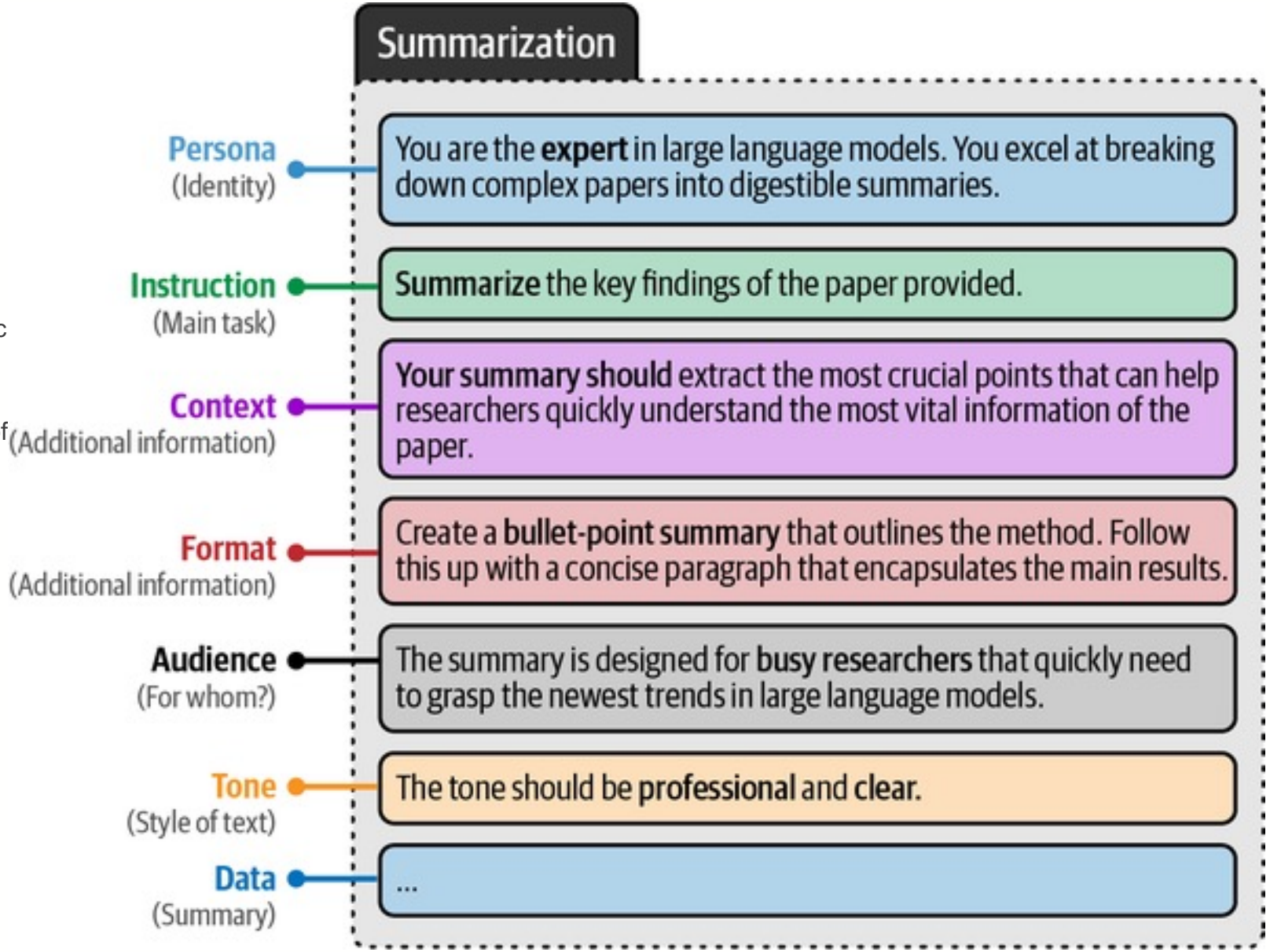
Context: Additional information describing the context of the problem or task. It answers questions like “What is the reason for the instruction?”

Format: The format the LLM should use to output the generated text. Without it, the LLM will come up with a format itself, which is troublesome in automated systems.

Audience: The target of the generated text. This also describes the level of the generated output. For education purposes, it is often helpful to use ELI5 (“Explain it like I’m 5”).

Tone: The tone of voice the LLM should use in the generated text. If you are writing a formal email to your boss, you might not want to use an informal tone of voice.

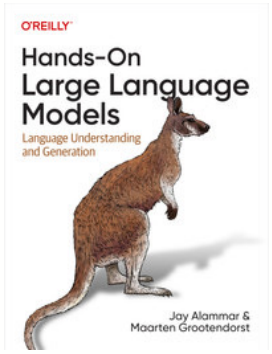
Data: The main data related to the task itself.



Providing Examples

Provide the LLM with **examples** of exactly the thing that we want to achieve.

Often referred to as **in-context learning**, where we provide the model with correct examples



Zero-shot prompt

Prompting without examples

Classify the text into neutral, negative, or positive.

Text: I think the food was okay.
Sentiment: ...

One-shot prompt

Prompting with a single example

Classify the text into neutral, negative, or positive.

Text: I think the food was alright.
Sentiment: **Neutral**

Text: I think the food was okay.
Sentiment:

Few-shot prompt

Prompting with more than one example

Classify the text into neutral, negative, or positive.

Text: I think the food was alright.
Sentiment: **Neutral.**

Text: I think the food was great!
Sentiment: **Positive.**

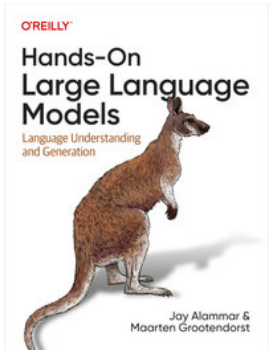
Text: I think the food was horrible...
Sentiment: **Negative.**

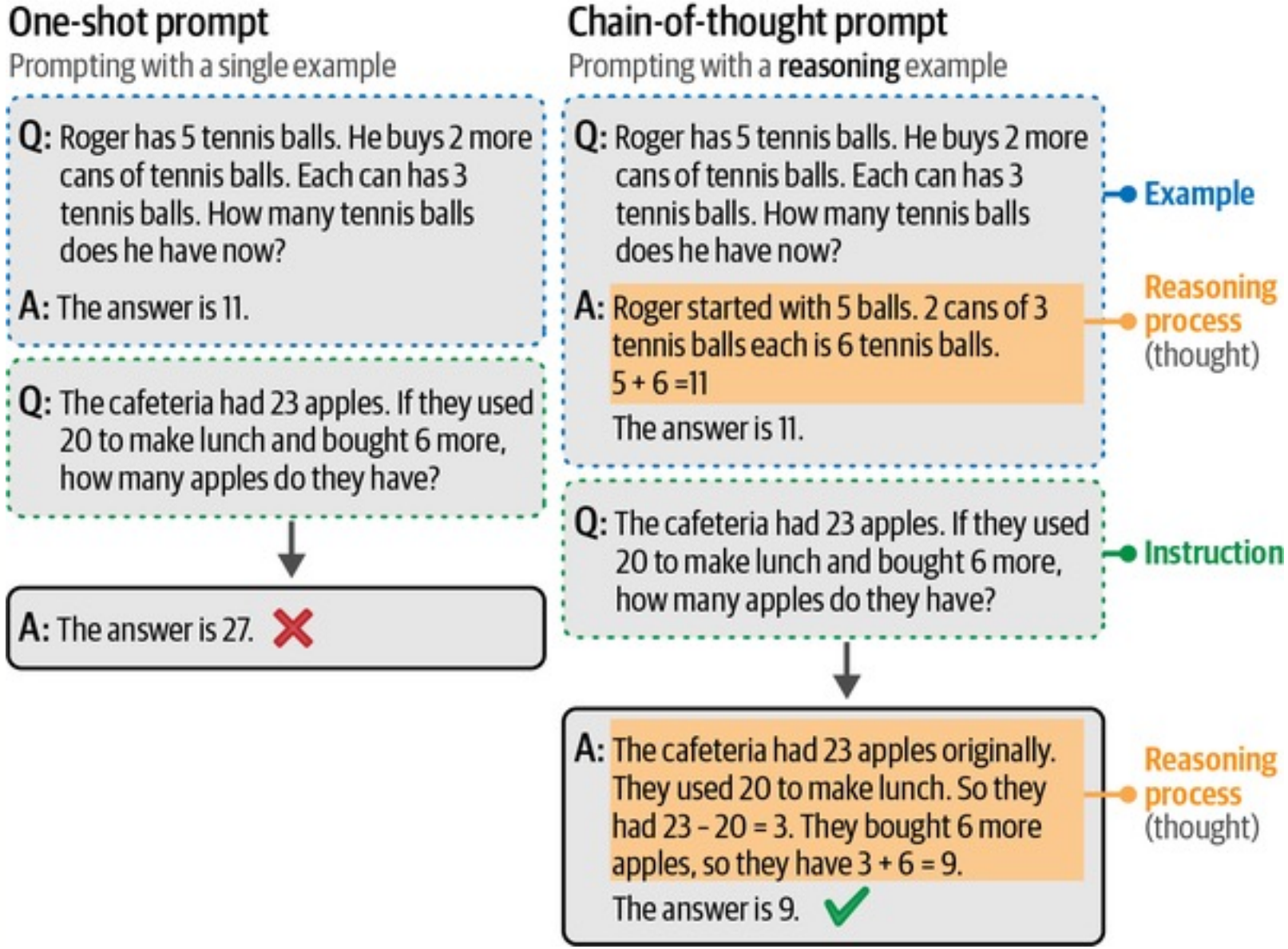
Text: I think the food was okay.
Sentiment:

Chain-of-Thought Think Before Answering

A first step toward **complex reasoning in generative models** was through a method called chain-of-thought.

Chain-of-thought aims to have the generative model “think” first rather than answering the question directly without any reasoning.





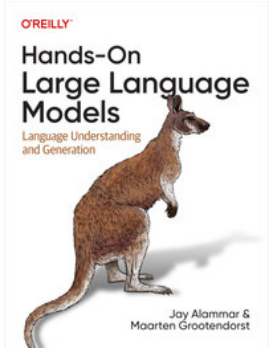
ZS-COT

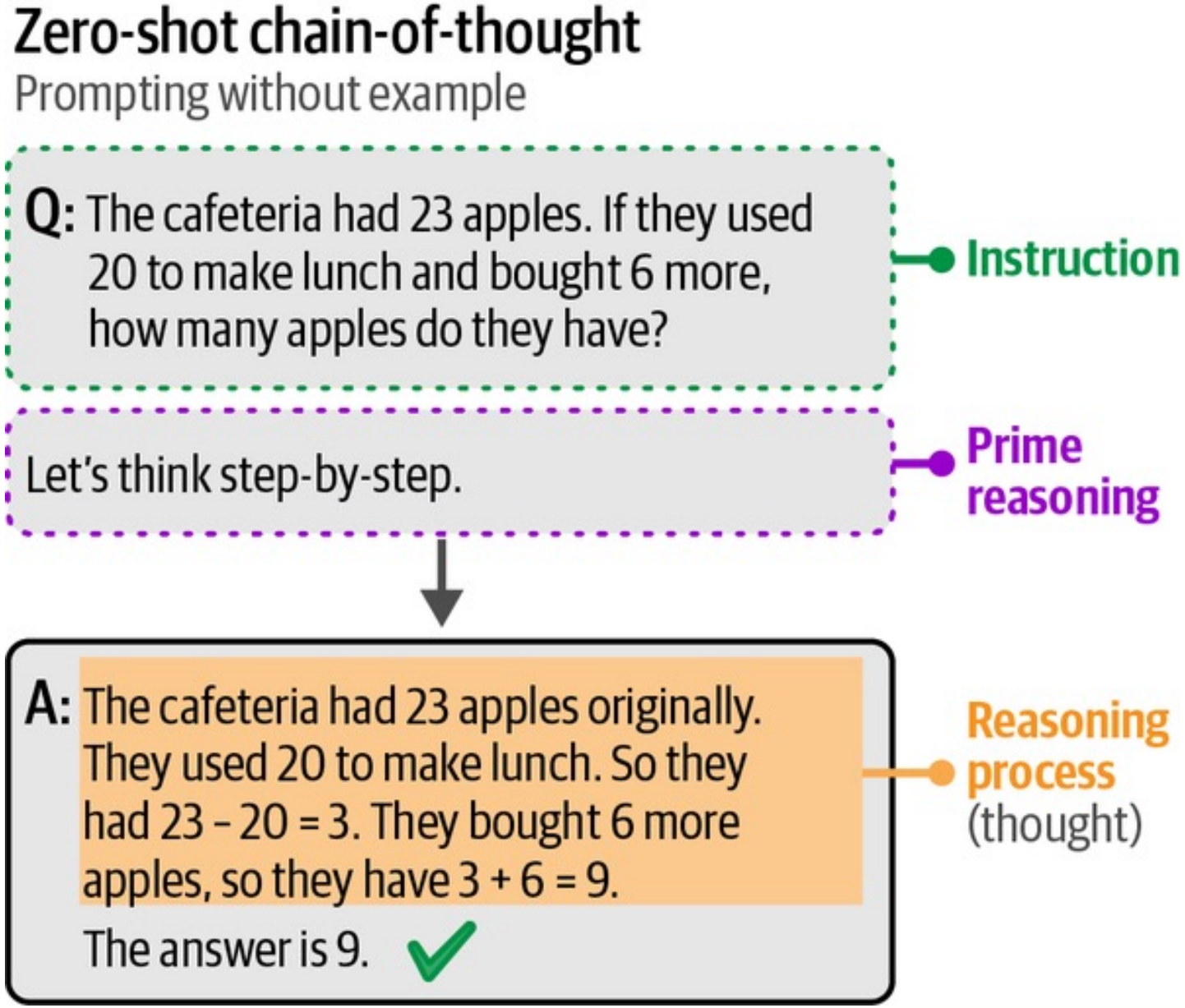
Prime reasoning:

"Let's think step-by-step"

"show your work"

https://learning.oreilly.com/library/view/hands-on-large-language/9781098150952/ch06.html





Self-Consistency

Variable Results

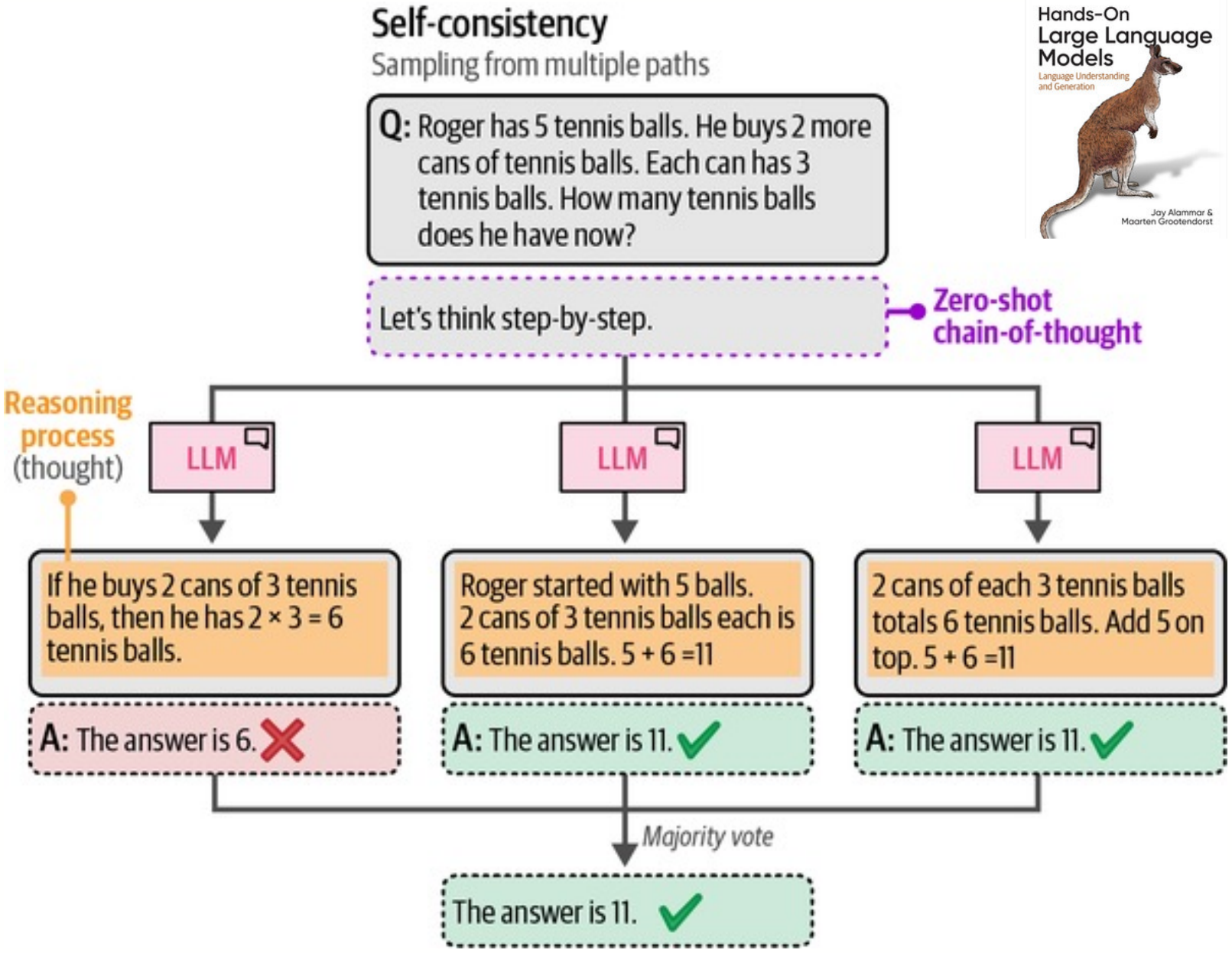
- Repeating prompts can yield different results
- Influenced by parameters like temperature and top_p
- Output quality may vary due to randomness

Self-Consistency Method

- Reduces randomness in generative models
- Uses majority result from multiple prompt responses
- Enhances performance and reliability

Diverse Sampling

- Each response can use different temperature and top_p values
- Increases diversity in model outputs



Tree of Thought

Complex Reasoning

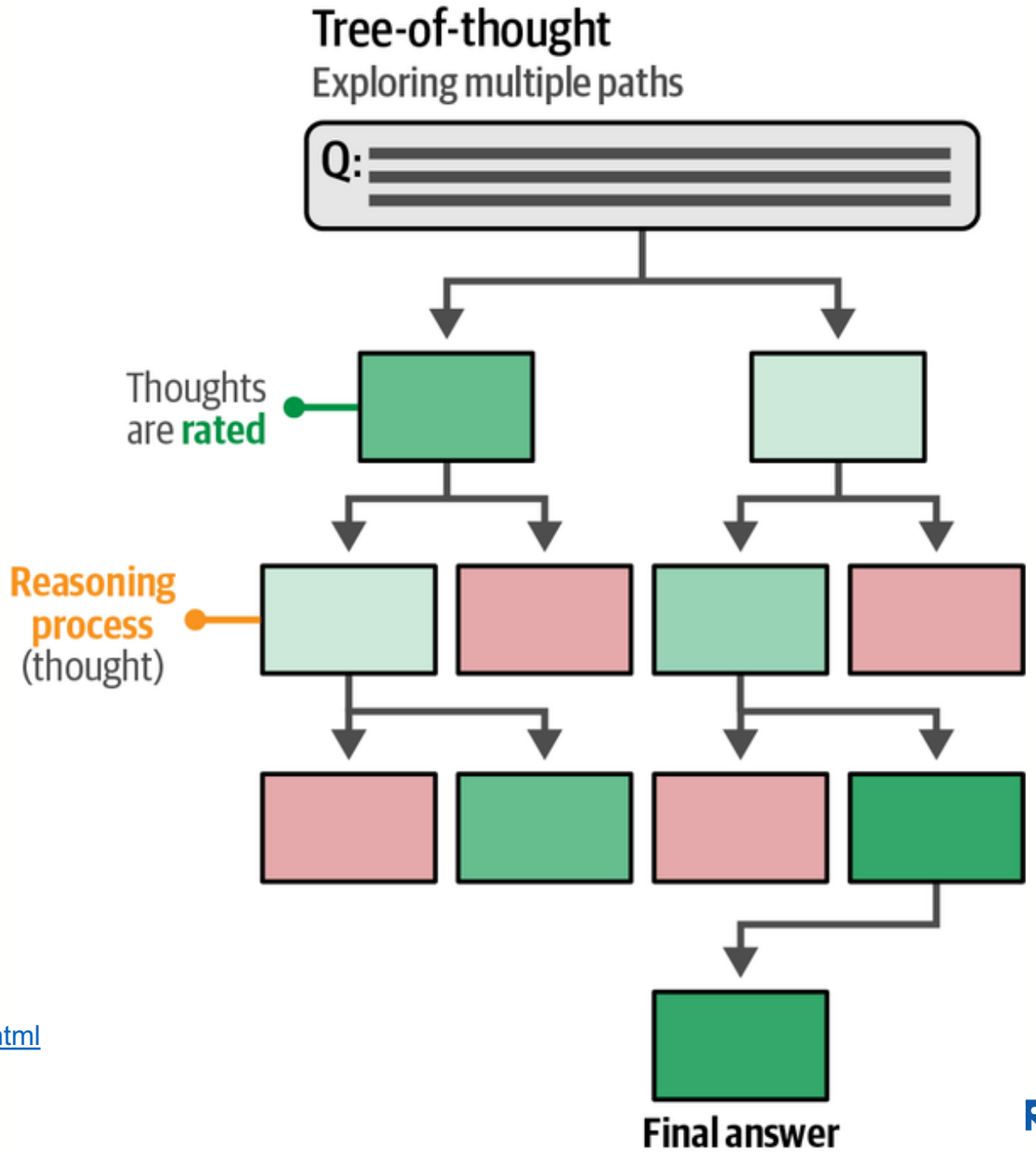
- Chain-of-thought and self-consistency enhance reasoning
- Samples multiple "thoughts" for improved model output

Advanced Techniques

- Current methods mimic complex reasoning
- Tree-of-thought offers deeper exploration of ideas

Problem-Solving Method

- Breaks problems into multiple reasoning steps
- Model explores solutions at each step
- Votes for the best solution before proceeding



https://learning.oreilly.com/library/view/hands-on-large-language/9781098150952/ch06.html

More Prompts

Computer Vision

"Generate a high-resolution image of a cat."

"Create a new artistic rendering of a landscape."

"Translate this sketch into a realistic image."

"Enhance the resolution of this low-quality photograph."

"Remove the background from this image while preserving the foreground object."

"Generate a cartoon version of this portrait."

"Modify the lighting conditions in this image."

"Apply a specific artistic style to this photograph."

"Colorize this black and white image."

"Generate a realistic image of a car from a textual description."

Multi-modal

“Generate a descriptive caption for this image."

"Create a video sequence based on this textual storyline."

"Translate this English text into a corresponding image."

"Generate an audio description for this visual scene."

"Produce a video with synchronized audio based on this script."

"Generate a textual summary of this video clip."

"Create a slideshow presentation from this text document."

"Translate this image into a sequence of musical notes."

"Generate a storyboard based on this audio narration."

"Create a comic strip from this dialogue script and accompanying images."

AI Agents

Image created with
Microsoft Copilot

Robinson

AI Agents

AI agents in generative AI, particularly within the context of Large Language Models (LLMs), refer to advanced AI systems that leverage the generative capabilities of these models to perform a wide range of tasks autonomously.

Human-like Interaction

- Understands and generates human-like text
- Interacts with users and comprehends instructions
- Executes tasks with complex reasoning and multi-step processes

Augmented Capabilities

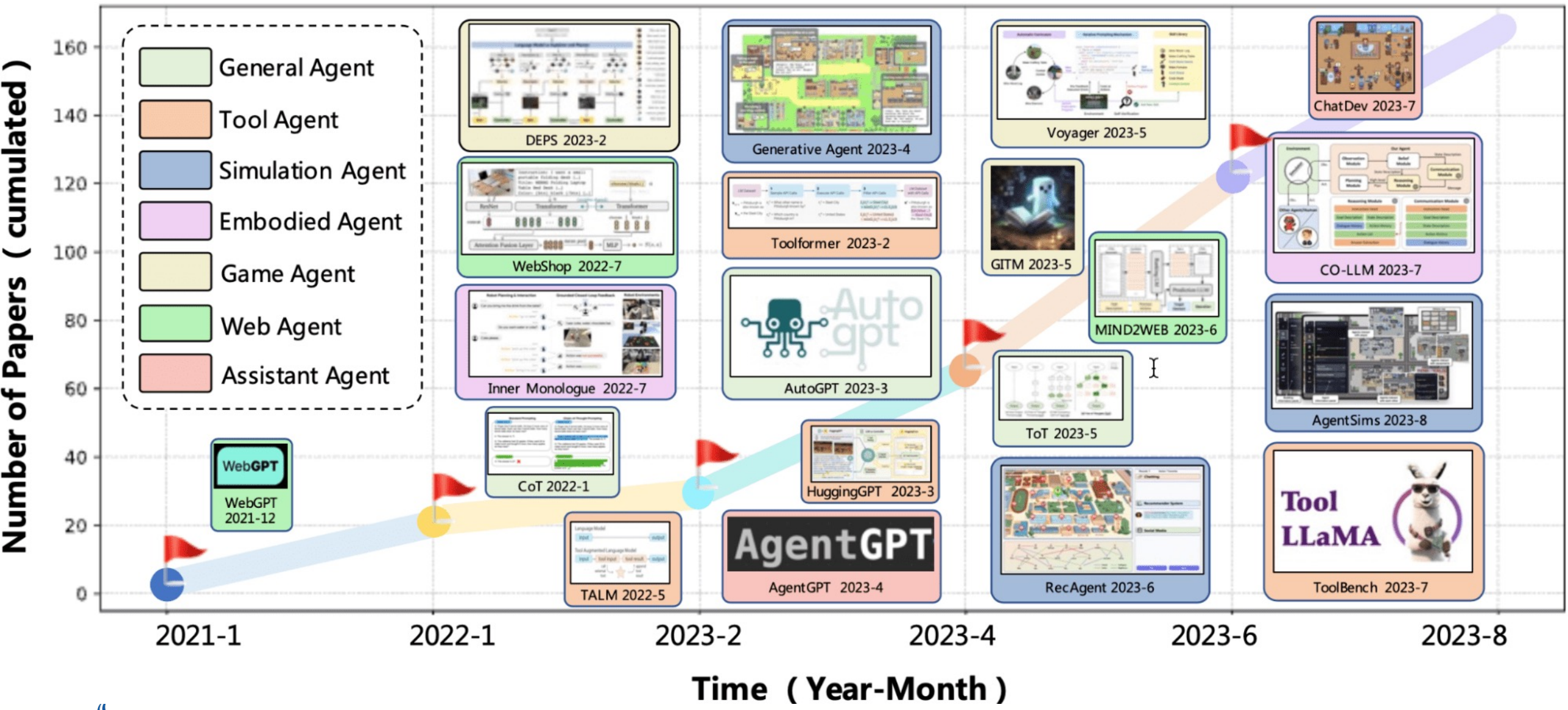
- Accesses external databases and uses APIs
- Incorporates updated information for dynamic environments
- Adapts to new challenges continuously

Foundation on LLMs

- Uses Large Language Models (LLMs) as a foundation
- Learns from examples and improves over time

Versatile and Intelligent Systems

- Acts as personal assistants or customer service agents
- Components of larger autonomous systems


General Agent
Tool Agent
Simulation Agent
Embodied Agent
Game Agent
Web Agent
Assistant Agent
Number of Papers (cumulated)
0
20
40
60
80
100
120
140
160
2021-1
2022-1
2023-2
2023-4
2023-6
2023-8
Time (Year-Month)
WebGPT 2021-12
DEPS 2023-2
WebShop 2022-7
Inner Monologue 2022-7
CoT 2022-1
TALM 2022-5
Generative Agent 2023-4
Toolformer 2023-2
AutoGPT 2023-3
HuggingGPT 2023-3
AgentGPT 2023-4
Voyager 2023-5
GITM 2023-5
MIND2WEB 2023-6
ToT 2023-5
RecAgent 2023-6
ChatDev 2023-7
CO-LLM 2023-7
AgentSims 2023-8
ToolBench 2023-7

Profile

Profile Contents

- Demographic Information
- Personality Information
- Social Information

Generation Strategy

- Handcrafting Method
- LLM-Generation Method
- Dataset Alignment Method

Memory

Memory Structure

- Unified Memory
- Hybrid Memory

Memory Formats

- Languages
- Databases
- Embeddings
- Lists

Memory Operation

- Memory Reading
- Memory Writing
- Memory Reflection

Planning

Planning w/o Feedback

- Single-path Reasoning
- Multi-path Reasoning
- External Planner

Planning w/ Feedback

- Environment Feedback
- Human Feedback
- Model Feedback

Action

Action Target

- Task Completion
- Exploration
- Communication

Action Production

- Memory Recollection
- Plan Following

Action Space

- Tools
- Self-Knowledge

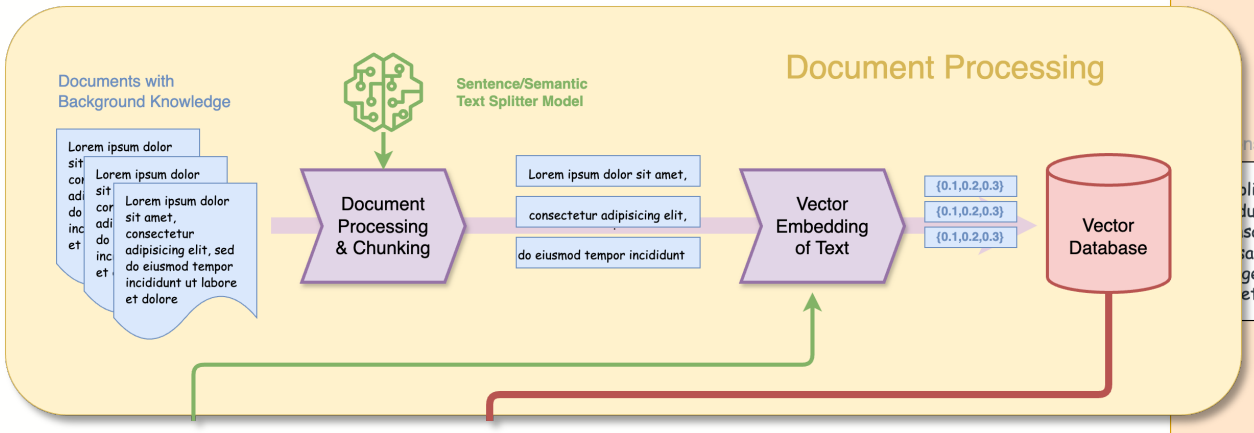
Action Impact

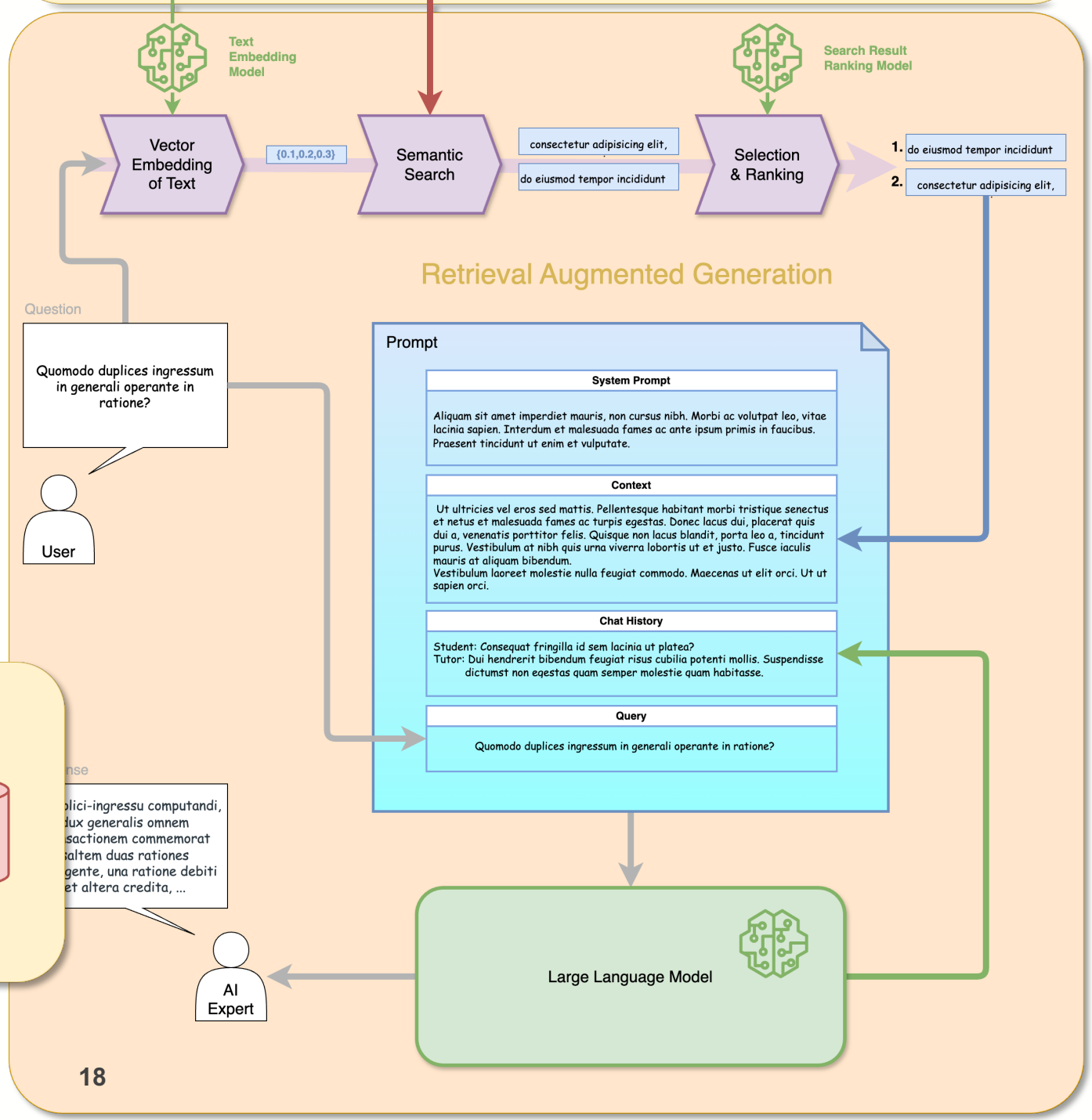
- Environments
- New Actions
- Internal States

Augmentation with External Knowledge (RAG)

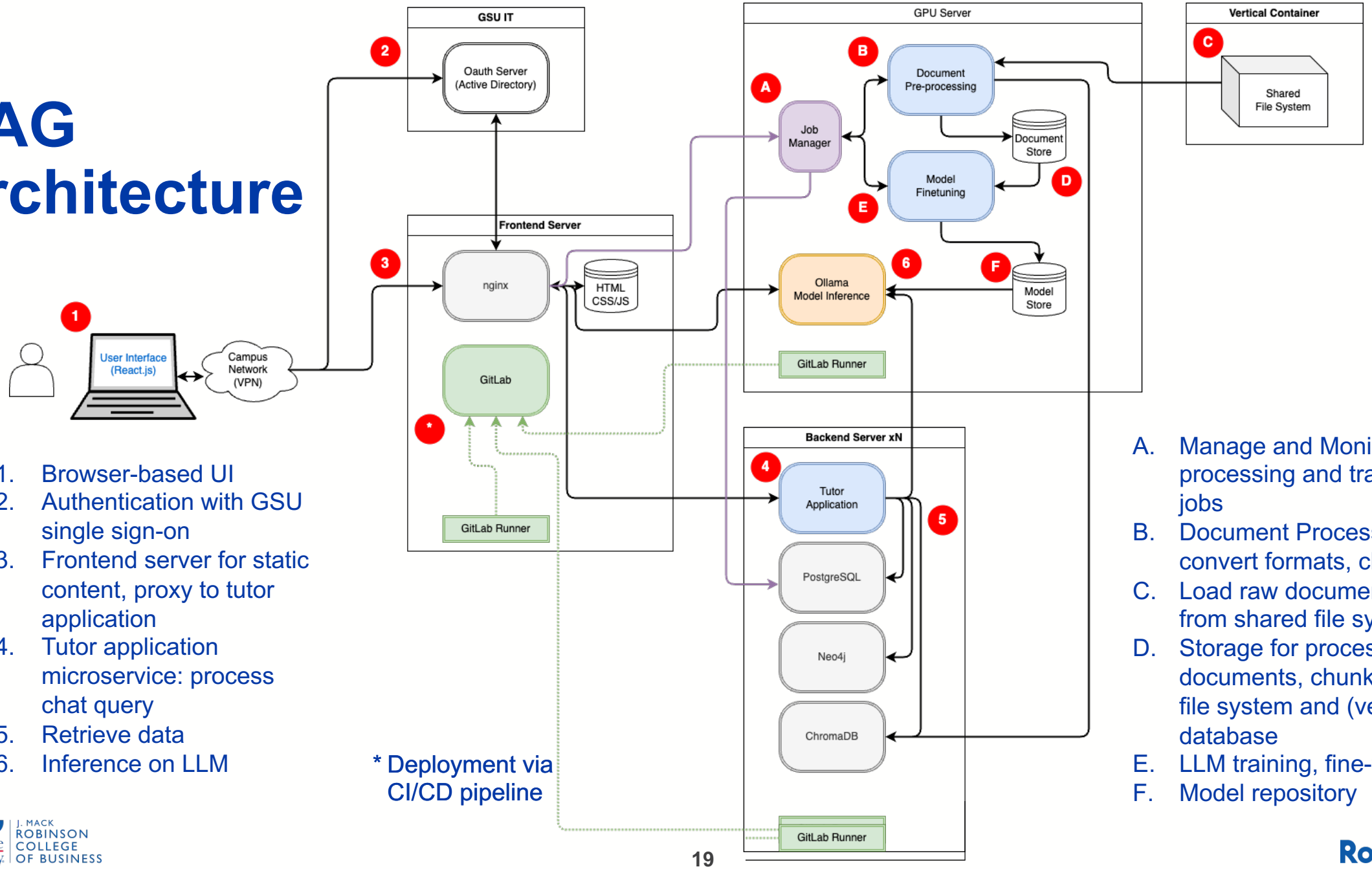
Retrieval Augmented Generation (RAG) enhances LLMs by **integrating external knowledge sources**, improving the model's ability to provide relevant and up-to-date responses.

RAG involves extracting queries from input prompts, **retrieving information from external sources, and incorporating this information into the model's responses**.





RAG Architecture



1. Browser-based UI
2. Authentication with GSU single sign-on
3. Frontend server for static content, proxy to tutor application
4. Tutor application microservice: process chat query
5. Retrieve data
6. Inference on LLM

* Deployment via CI/CD pipeline

A. Manage and Monitor processing and training jobs
B. Document Processing: convert formats, chunking
C. Load raw documents from shared file system
D. Storage for processed documents, chunks on file system and (vector) database
E. LLM training, fine-tuning
F. Model repository

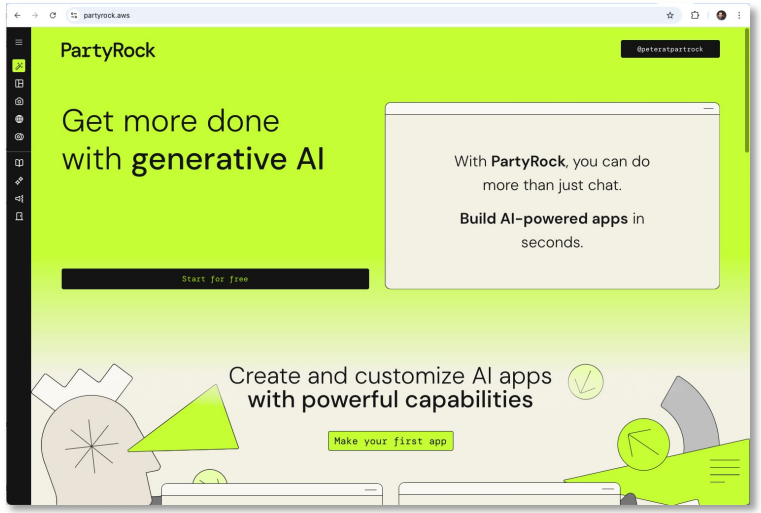
Activities

Image created with
Microsoft Copilot

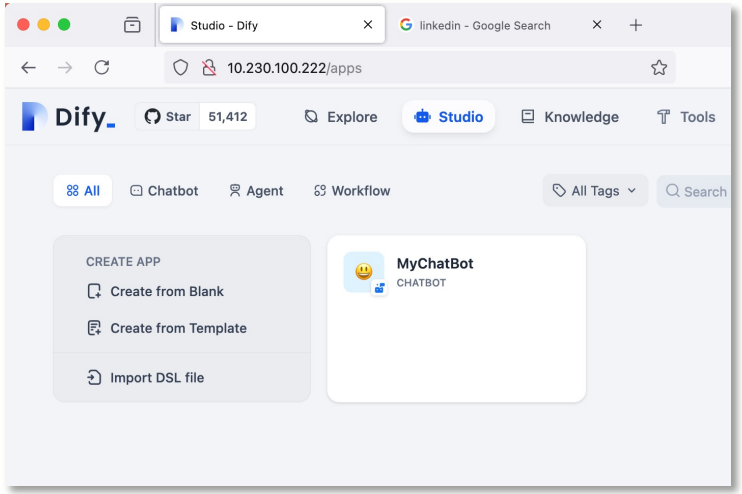
Robinson

Tools for todays Activities

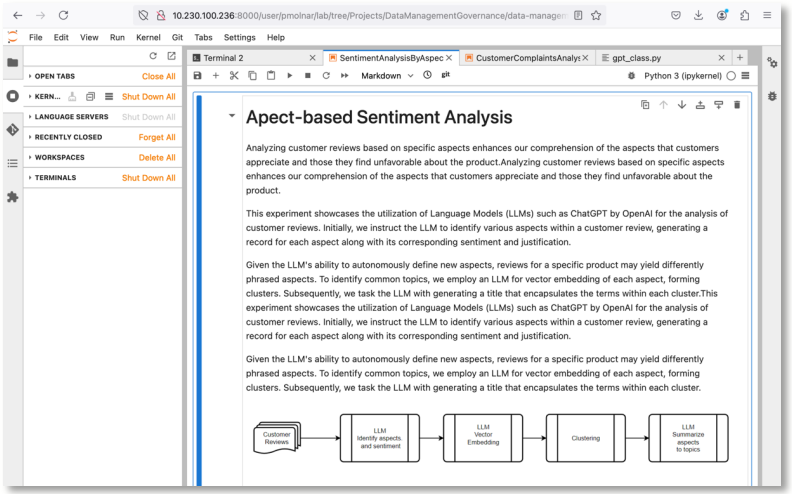
AWS PartyRock



DIFY (self hosted)



Python/Jupyter Notebook



What do you want to build?

I want to build an app that approves load applications based in the applicants credit score, monthly income and loan amount. If the credit score is less than 600 the loan is approved if it's less than 3 times the monthly income. If the credit score is between 600 and 720, the approved load amount is 5 time the monthly income. If the score is above 720 the load is approved up to 7 times the monthly income.

Inspire me with example prompts

Credit-Wise Lending Advisor

Snapshot Remix Share Edit

Loan Approval Calculator

Welcome to the Loan Approval Calculator. This tool will determine if your loan application is approved based on your credit score, monthly income, and requested loan amount. Please enter your information in the fields below.

Credit Score

→ Enter your credit score (300-850)

Monthly Income

→ Enter your monthly income in dollars

Requested Loan Amount

→ Enter the loan amount you're requesting

Loan Approval Result

Fill in Credit Score, Monthly Income and Requested Loan Amount to get started!

Play the app with this button or use Cmd + Enter. You can enable autoplay from the top right corner.

Cmd + Enter to play app

partyrock.aws/u/peteratpartrock/JXre633Kk/Credit-Wise-Lending-Advisor

Credit-Wise Lending Advisor

+ Add widget | Leave edit

Welcome to the Loan Approval Calculator. This tool will determine if your loan application is approved based on your credit score, monthly income, and requested loan amount. Please enter your information in the fields below.

Credit Score	Monthly Income	Requested Loan Amount
590	2000	4000

Loan Approval Result

First, we need to determine which rule applies based on the credit score:

The credit score is 590, which is less than 600. Therefore, we will use Rule 1: "If credit score < 600, approve if loan amount is < 3 times monthly income"

Now, let's calculate and compare:

1. Maximum loan amount according to Rule 1: 3 x Monthly Income = 3 x $2000 = $6000
2. Requested loan amount: $4000
3. Comparison: $4000 (requested) < $6000 (maximum allowed)

Decision: The loan is APPROVED.

Explanation: The applicant's credit score of 590 falls into the lowest category (< 600). For this category, we approve loans that are less than 3 times the monthly income. The maximum loan amount for this applicant would be $6000 (3 x $2000).

The requested loan amount of $4000 is less than the maximum allowed amount of $6000. Therefore, the loan application meets the criteria for approval.

Maximum loan amount: $6000

Prompt | Labels | Model

Prompt

Press @ to reference another widget.

Based on the following information:

Credit Score: Credit Score
Monthly Income: $ Monthly Income
Requested Loan Amount: $
Requested Loan Amount

Determine if the loan is approved using these rules:
1. If credit score < 600, approve if loan amount is < 3 times monthly income
2. If 600 <= credit score < 720, approve if loan amount is <= 5 times monthly income
3. If credit score >= 720, approve if loan amount is <= 7 times monthly income

Provide a detailed explanation of the decision, including calculations and comparisons. If approved, state the maximum loan amount. If not approved, explain why and what the maximum approved amount would be.

Delete widget | Cancel | Save

partyrock.aws/u/peteratpartrock/JXre633Kk/Credit-Wise-Lending-Advisor

Credit-Wise Lending Advisor

+ Add widget | Leave edit

Credit Score

767

Monthly Income

3000

Requested Loan Amount

10000

Loan Approval Result

3. Requested Loan Amount: $10000

For credit scores >= 720, the rule states that the loan is approved if the loan amount is <= 7 times monthly income.

Calculation: Maximum loan amount = 7 * Monthly Income = 7 * $3000 = $21000

Comparison: Requested loan amount ($10000) <= Maximum loan amount ($21000)

Decision: The loan is APPROVED.

Explanation: With a credit score of 767, the applicant falls into the highest credit category. For this category, loans are approved if the requested amount is less than or equal to 7 times the monthly income.

The maximum loan amount this applicant could be approved for is $21000 (7 * $3000).

The requested loan amount of $10000 is well below this maximum. It is actually only 3.33 times the monthly income ($10000 / $3000 = 3.33), which is well within the allowed limit for this credit score category.

Therefore, based on the given rules and the applicant's financial information, the loan of $10000 is approved. The applicant could potentially be approved for a loan of up to $21000 if they needed it.

widget6 PROMPT

Editing

Loan Advisor Chat

Hello! I'm here to help you with any questions you might have about your loan application or the approval process. Feel free to ask me anything!

Prompt | Labels | Model

Image description

Press @ to reference another widget.

An image of Loan Approval Result. Draw a smiling happy teddy bear if the decision is APPROVED. Otherwise, draw a frowning sad teddy bear.

Delete widget | Cancel | Save

Credit-Wise Lending Advisor

+ Add widget | Leave edit

loan amount is <= 7 times the monthly income.

Calculation: Maximum allowable loan amount = 7 × Monthly Income Maximum allowable loan amount = 7 × $3,000 = $21,000

Comparison: Requested loan amount ($13,000) < Maximum allowable loan amount ($21,000)

Decision: The loan is APPROVED.

Explanation: With a credit score of 767, the applicant falls into the highest credit category. For this category, loans up to 7 times the monthly income can be approved.

The maximum loan amount this applicant could be approved for is $21,000 (7 × $3,000).

The requested loan amount of $13,000 is well within this limit, being only 4.33

Loan Advisor Chat

Hello! I'm here to help you with any questions you might have about your loan application or the approval process. Feel free to ask me anything!

Prompt | Labels | Model

Prompt

Press @ to reference another widget.

The user has entered their credit score (Credit Score), monthly income ($ Monthly Income), and requested loan amount ($ Requested Loan Amount). They may have questions about their loan approval status or the approval process.

Delete widget | Cancel | Save

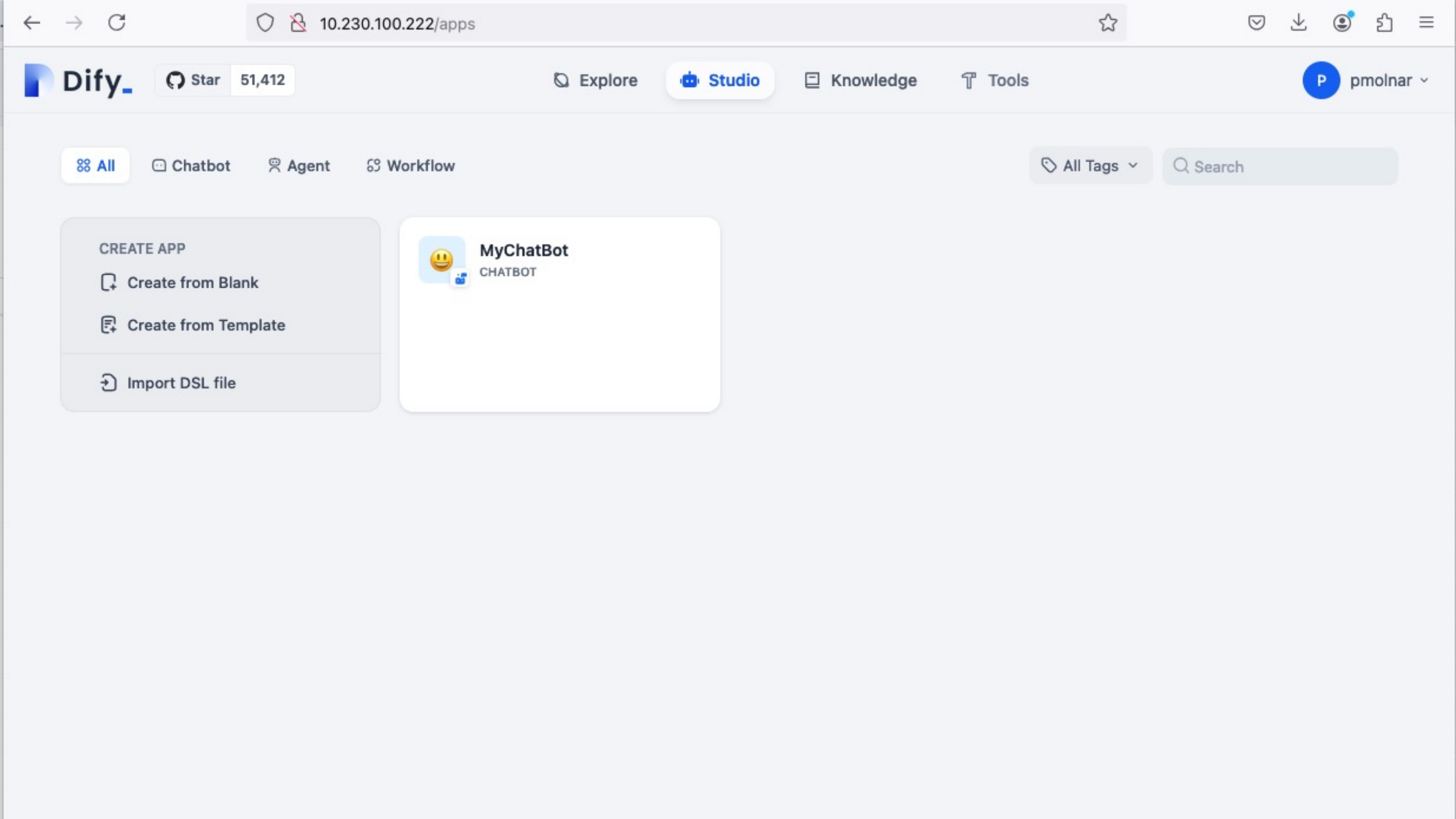
Getting started with AWS PartyRock

1. Visit https://partyrock.aws
2. Select "Get started for free"
3. Sign in with your personal Google, Apple or Amazon account
4. Setup your username

The system does NOT REQUIRE a credit card.

No charges, Just a usage limit.

Make sure to clear your browser cache and data after this session!


10.230.100.222/apps
Dify_
Star 51,412
Explore
Studio
Knowledge
Tools
P
pmolnar
All
Chatbot
Agent
Workflow
All Tags
Search
CREATE APP
Create from Blank
Create from Template
Import DSL file
MyChatBot
CHATBOT

MyChatBot
CHATBOT BASIC

Orchestrate
API Access
Logs & Ann.
Monitoring

Orchestrate

gpt-4o-audio-preview CHAT | Publish

Instructions

Generate

Write your prompt word here, enter '{' to insert a variable, enter '/' ...

0

Variables

+ Add

Variables allow users to introduce prompt words or opening remarks when filling out forms. You can try entering "{{input}}" in the prompt words.

Context

Retrieval Setting + Add

You can import Knowledge as context

Vision

Settings

Debug & Preview

Talk to Bot

Features Enabled Manage →

GenAI_regulation
--

Documents
Retrieval Testing
Settings

The Knowledge has not been associated, please go to the application or plug-in to complete the association.

View documentation

GenAI_Report_Florence-GSELL.pdf

Available | Add chunk

1,244 Paragraphs

All | Search

001 Enabled

FLORENCE G'SELL REGULATING UNDER UNCERTAINTY: Governance Options for Generative AI

002 Enabled

Acknowledgments 1 Executive Summary 2 Chapter 1: Introduction 8 Chapter 2: Generative AI: The technology and supply chain 29 Chapter 3: Challenges and risks of generative AI 58 Chapter 4: Industry

004 Enabled

In particular, Professor Jingwen Wang and Professor Xinyu Fu provided invaluable insights on generative AI technology, while Dave Willner shared his expertise on the generative AI industry. This work has also benefited

005 Enabled

2 The revolution underway in the development of artificial intelligence promises to transform the economy and all social systems. It is difficult to think of an area of life that will not be affected in some way by AI, if the

007 Enabled

3 require government action.

008 Enabled

It will have to ascertain if a generative

Metadata

Labeling metadata for documents allows AI to access them in a timely manner and exposes the source of references for users.

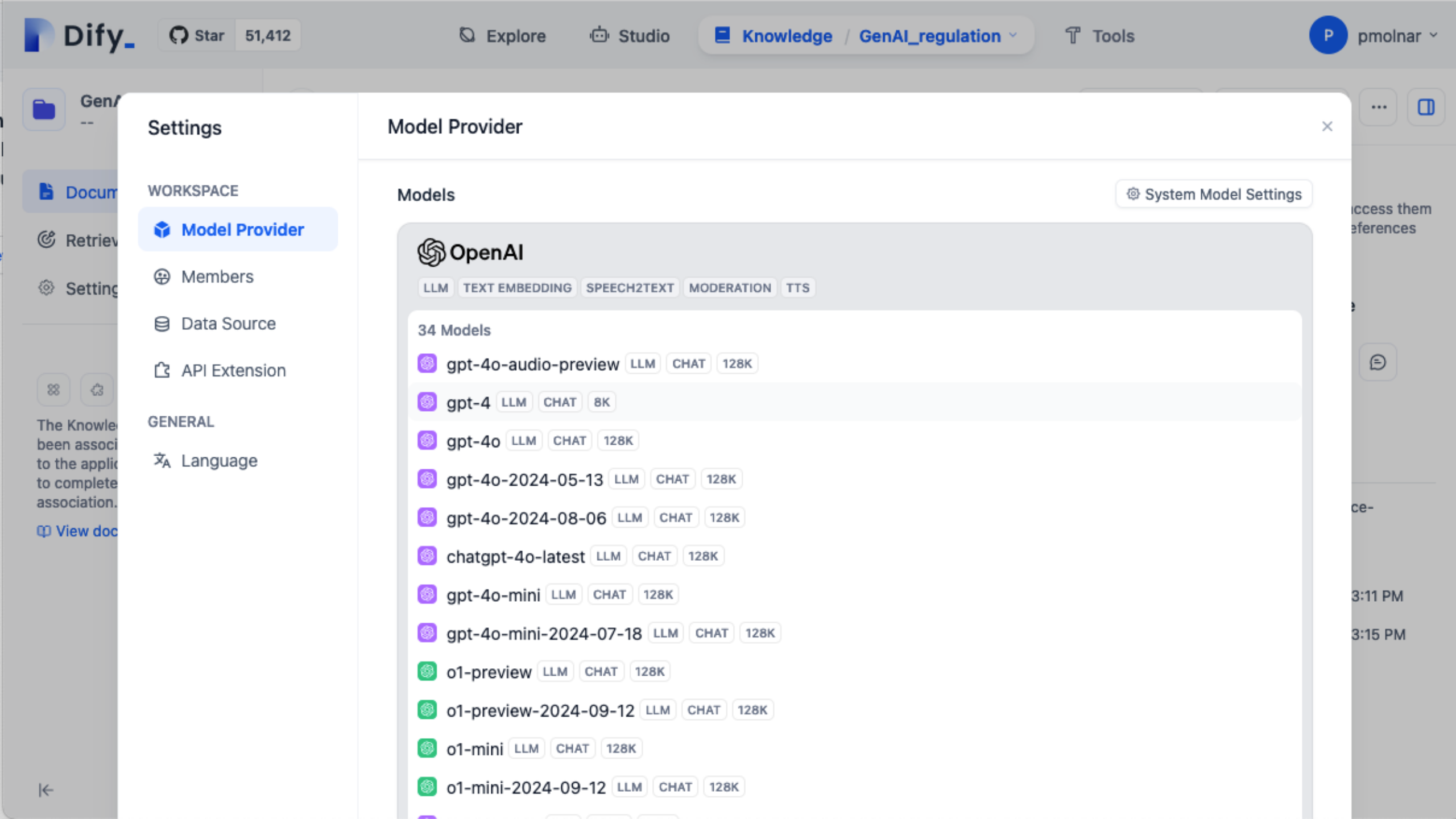
Please select a document type

Let's go

Original filename	GenAI_Report_Florence-GSELL.pdf
Original file size	9.88MB
Upload date	November 12, 2024 03:11 PM
Last update date	November 12, 2024 03:15 PM
Source	Upload File

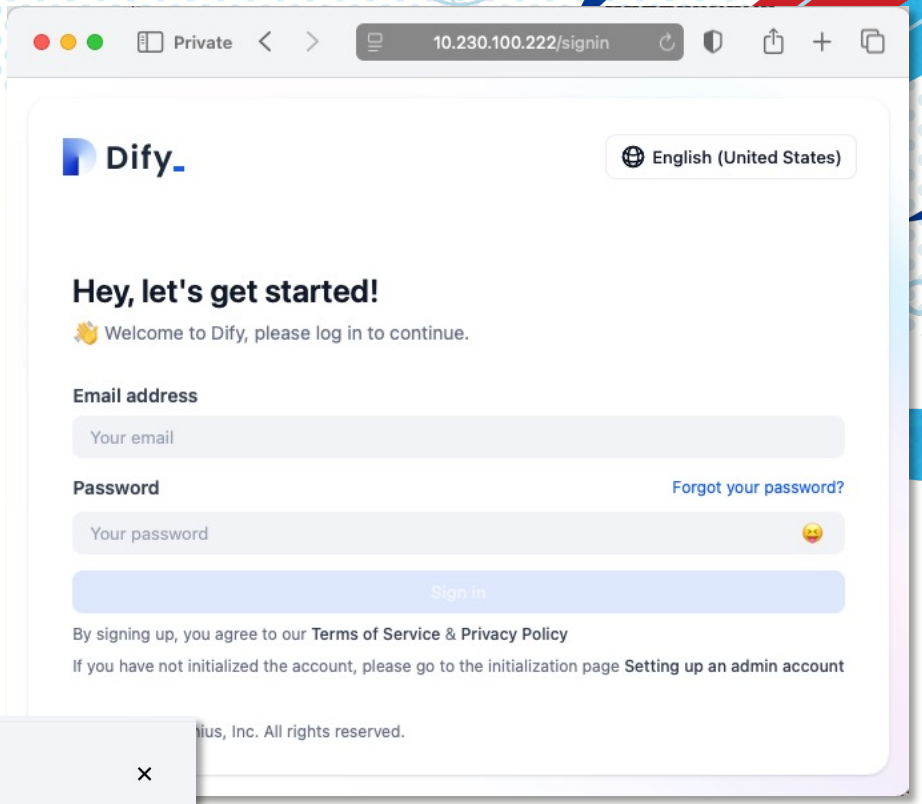
Technical Parameters

Chunks specification	Automatic

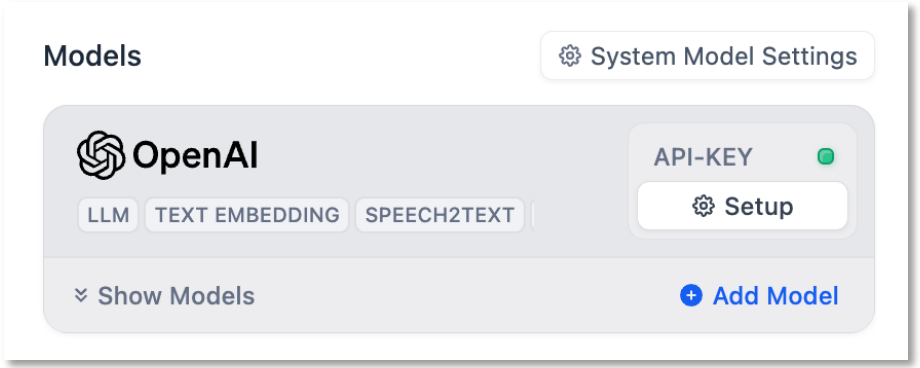

Dify
Star 51,412
Explore
Studio
Knowledge / GenAI_regulation
Tools
pmolnar
Settings
WORKSPACE
Model Provider
Members
Data Source
API Extension
GENERAL
Language
Model Provider
Models
System Model Settings
OpenAI
LLM TEXT EMBEDDING SPEECH2TEXT MODERATION TTS
34 Models
gpt-4o-audio-preview LLM CHAT 128K
gpt-4 LLM CHAT 8K
gpt-4o LLM CHAT 128K
gpt-4o-2024-05-13 LLM CHAT 128K
gpt-4o-2024-08-06 LLM CHAT 128K
chatgpt-4o-latest LLM CHAT 128K
gpt-4o-mini LLM CHAT 128K
gpt-4o-mini-2024-07-18 LLM CHAT 128K
o1-preview LLM CHAT 128K
o1-preview-2024-09-12 LLM CHAT 128K
o1-mini LLM CHAT 128K
o1-mini-2024-09-12 LLM CHAT 128K

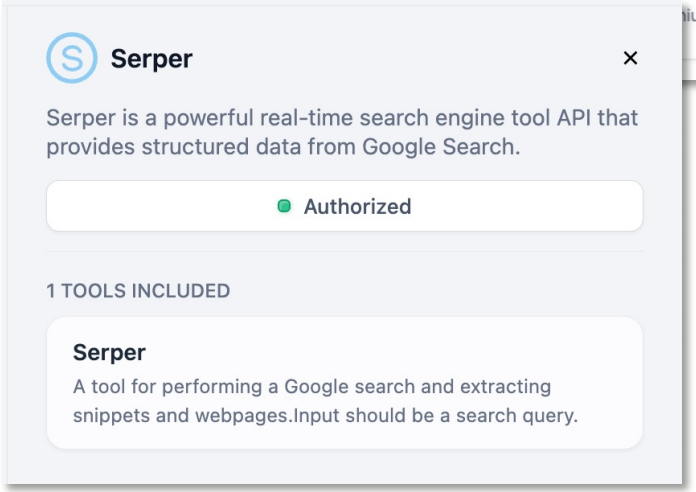
Getting started with DIFY

1. Visit http://10.230.100.222/
2. Sign in with one of the guest accounts (use the guest email)



API Access granted to:





Getting started with Jupyter

1. Visit http://10.230.100.236:8000/
2. Log in with one of the guest accounts
3. Clone GIT repository: https://github.com/molnarai/genai-exercises.git
4. Choose "Conda Python 3.12" kernel

Packages:

Langchain, Chromadb, Ollama

(additional packages can be installed)