

NirDiamant /
GenAI_Agents

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

This repository provides tutorials and implementations for various Generative AI Agent techniques, from basic to advanced. It serves as a comprehensive guide for building intelligent, interactive AI systems.

View license

☆ 3.4k stars

🔗 345 forks

👁 52 watching

🔗

📁

📈 Activity

Branches

Tags

🌐

 Public repository

1 Branch

0 Tags

🔍 Go to file

t


Go to file

+

Add file

<> Code

...

 **NirDiamant** updated README

4b4146d · 1 hour ago

| | | |
|------------------------|---|--------------|
| 📁 .github | copying files from private repo | last month |
| 📁 all_agents_tutorials | Merge branch 'main' of https://github... | 3 days ago |
| 📁 audio | updated tts | last month |
| 📁 images | added hackathon banner | last week |
| 📄 .gitignore | copying files from private repo | last month |
| 📄 CONTRIBUTING.md | updated contributing | 13 hours ago |
| 📄 LICENSE | updated license | last week |
| 📄 README.md | updated README | 1 hour ago |
| 📄 requirements.txt | chore: upgrade duckduckgo to late... | 2 weeks ago |

📖 README

📄 License

PRs welcome

LinkedIn Connect

Follow @NirDiamantAI

Discord

Join our community

🌟 Support This Project:

Your sponsorship fuels innovation in GenAI agent development. [Become a sponsor](#) to help maintain and expand this valuable resource!

GenAI Agents: Comprehensive Repository for Development and Implementation

Welcome to one of the most extensive and dynamic collections of Generative AI (GenAI) agent tutorials and implementations available today. This repository serves as a comprehensive resource for learning, building, and sharing GenAI agents, ranging from simple conversational bots to complex, multi-agent systems.

Stay Updated!

Don't miss out on cutting-edge developments, new tutorials, and community insights!

[Subscribe to DiamantAI's Newsletter](#)

Join the AgentCraft Hackathon in conjunction with LangChain!



[Register Now for the AgentCraft Hackathon!](#)

Join us to build innovative AI agents and contribute to this open source while winning prizes!

Introduction

Generative AI agents are at the forefront of artificial intelligence, revolutionizing the way we interact with and leverage AI technologies. This repository is designed to guide you through the development journey, from basic agent implementations to advanced, cutting-edge systems.

Our goal is to provide a valuable resource for everyone - from beginners taking their first steps in AI to seasoned practitioners pushing the boundaries of what's possible. By offering a range of examples from foundational to complex, we aim to facilitate learning, experimentation, and innovation in the rapidly evolving field of GenAI agents.

Furthermore, this repository serves as a platform for showcasing innovative agent creations. Whether you've developed a novel agent architecture or found an innovative application for existing techniques, we encourage you to share your work with the community.

Related Projects

📖 Dive into my [comprehensive guide on RAG techniques](#) to learn about integrating external knowledge into AI systems, enhancing their capabilities with up-to-date and relevant information retrieval.

✍️ Explore my [Prompt Engineering Techniques guide](#) for an extensive collection of prompting strategies, from fundamental concepts to advanced methods, improving your ability to communicate effectively with AI language models.

A Community-Driven Knowledge Hub

This repository grows stronger with your contributions! Join our vibrant Discord community — the central hub for shaping and advancing this project together 🤝

[GenAI Agents Discord Community](#)

Whether you're a novice eager to learn or an expert ready to share your knowledge, your insights can shape the future of GenAI agents. Join us to propose ideas, get feedback, and collaborate on innovative implementations. For contribution guidelines, please refer to our [CONTRIBUTING.md](#) file. Let's advance GenAI agent technology together!

🔗 For discussions on GenAI, agents, or to explore knowledge-sharing opportunities, feel free to [connect on LinkedIn](#).

Key Features

- 🎓 Learn to build GenAI agents from beginner to advanced levels
- 🧠 Explore a wide range of agent architectures and applications
- 📖 Step-by-step tutorials and comprehensive documentation
- 🛠️ Practical, ready-to-use agent implementations
- 🌟 Regular updates with the latest advancements in GenAI
- 🤝 Share your own agent creations with the community

GenAI Agent Implementations

Explore our extensive list of GenAI agent implementations, ranging from simple to complex:

🌱 Beginner-Friendly Agents

1. [Simple Conversational Agent](#)

Overview 🔍

A context-aware conversational AI maintains information across interactions, enabling more natural dialogues.

Implementation 🛠️

Integrates a language model, prompt template, and history manager to generate contextual responses and track conversation sessions.

2. [Simple Question Answering Agent](#)

Overview 🔍

Answering (QA) agent using LangChain and OpenAI's language model understands user queries and provides relevant, concise answers.

Implementation 🛠️

Combines OpenAI's GPT model, a prompt template, and an LLMChain to process user questions and generate AI-driven responses in a streamlined manner.

3. [Simple Data Analysis Agent](#)

Overview 🔍

An AI-powered data analysis agent interprets and answers questions about datasets using natural language, combining language models with data manipulation tools for intuitive data exploration.

Implementation 🛠️

Integrates a language model, data manipulation framework, and agent framework to process natural language queries and perform data analysis on a synthetic dataset, enabling accessible insights for non-technical users.

🛠️ Framework Tutorial: LangGraph

4. [Introduction to LangGraph: Building Modular AI Workflows](#)

Overview 🔍

This tutorial introduces LangGraph, a powerful framework for creating modular, graph-based AI workflows. Learn how to leverage LangGraph to build more complex and flexible AI agents that can handle multi-step processes efficiently.

Implementation 🛠️

Step-by-step guide on using LangGraph to create a StateGraph workflow. The tutorial covers key concepts such as state management, node creation, and graph compilation. It demonstrates these principles by constructing a simple text analysis pipeline, serving as a foundation for more advanced agent architectures.

🔍 Task-Specific Agents

5. [Customer Support Agent \(LangGraph\)](#)

Overview 🔍

An intelligent customer support agent using LangGraph categorizes queries, analyzes sentiment, and provides appropriate responses or escalates issues, automating initial stages of customer interaction for improved efficiency and satisfaction.

Implementation

Utilizes LangGraph to create a workflow combining state management, query categorization, sentiment analysis, and response generation. The system uses TypedDict for state management and implements conditional routing based on query characteristics.

6. [Essay Grading Agent \(LangGraph\)](#)

Overview

An automated essay grading system using LangGraph and an LLM model evaluates essays based on relevance, grammar, structure, and depth of analysis, streamlining assessment processes in educational settings.

Implementation

Utilizes a state graph to define the grading workflow, incorporating separate grading functions for each criterion. The system employs conditional logic to determine the flow of the grading process based on interim scores, with a final weighted average calculation.

7. [Travel Planning Agent \(LangGraph\)](#)

Overview

A Travel Planner using LangGraph demonstrates how to build a stateful, multi-step conversational AI application that collects user input and generates personalized travel itineraries.

Implementation

Utilizes StateGraph to define the application flow, incorporates custom PlannerState for process management, and employs node functions for city input, interests input, and itinerary creation. The system integrates an LLM to generate the final personalized travel itinerary.

Creative and Generative Agents

8. [GIF Animation Generator Agent \(LangGraph\)](#)

Overview

A GIF animation generator that integrates LangGraph for workflow management, GPT-4 for text generation, and DALL-E for image creation, producing custom animations from user prompts.

Implementation

Utilizes LangGraph to orchestrate a workflow that generates character descriptions, plots, and image prompts using GPT-4, creates images with DALL-E 3, and assembles them into GIFs using PIL. Employs asynchronous programming for efficient parallel processing.

9. [TTS Poem Generator Agent \(LangGraph\)](#)

Overview 🔍

An advanced text-to-speech (TTS) agent using LangGraph and OpenAI's APIs classifies input text, processes it based on content type, and generates corresponding speech output.

Implementation 🛠️

Utilizes LangGraph to orchestrate a workflow that classifies input text using GPT models, applies content-specific processing, and converts the processed text to speech using OpenAI's TTS API. The system adapts its output based on the identified content type (general, poem, news, or joke).

10. [Music Composer Agent \(LangGraph\)](#)

Overview 🔍

An AI Music Composer using LangGraph and OpenAI's language models generates custom musical compositions based on user input. The system processes the input through specialized components, each contributing to the final musical piece, which is then converted to a playable MIDI file.

Implementation 🛠️

LangGraph orchestrates a workflow that transforms user input into a musical composition, using ChatOpenAI (GPT-4) to generate melody, harmony, and rhythm, which are then style-adapted. The final AI-generated composition is converted to a MIDI file using music21 and can be played back using pygame.

🚀 Advanced Agent Architectures

11. [Memory-Enhanced Conversational Agent](#)

Overview 🔍

A memory-enhanced conversational AI agent incorporates short-term and long-term memory systems to maintain context within conversations and across multiple sessions, improving interaction quality and personalization.

Implementation 🛠️

Integrates a language model with separate short-term and long-term memory stores, utilizes a prompt template incorporating both memory types, and employs a memory manager for storage and retrieval. The system includes an interaction loop that updates and utilizes memories for each response.

12. [Multi-Agent Collaboration System](#)

Overview 🔍

A multi-agent collaboration system combining historical research with data analysis, leveraging large language models to simulate specialized agents working together to answer complex historical questions.

Implementation 🛠️

Utilizes a base Agent class to create specialized HistoryResearchAgent and DataAnalysisAgent, orchestrated by a HistoryDataCollaborationSystem. The system follows a five-step process: historical context provision, data needs identification, historical data provision, data analysis, and final synthesis.

13. [Self-Improving Agent](#)

Overview 🔍

A Self-Improving Agent using LangChain engages in conversations, learns from interactions, and continuously improves its performance over time through reflection and adaptation.

Implementation 🛠️

Integrates a language model with chat history management, response generation, and a reflection mechanism. The system employs a learning system that incorporates insights from reflection to enhance future performance, creating a continuous improvement loop.

14. [Task-Oriented Agent](#)

Overview 🔍

A language model application using LangChain that summarizes text and translates the summary to Spanish, combining custom functions, structured tools, and an agent for efficient text processing.

Implementation 🛠️

Utilizes custom functions for summarization and translation, wrapped as structured tools. Employs a prompt template to guide the agent, which orchestrates the use of tools. An agent executor manages the process, taking input text and producing both an English summary and its Spanish translation.

15. [Internet Search and Summarize Agent](#)

Overview 🔍

An intelligent web research assistant that combines web search capabilities with AI-powered summarization, automating the process of gathering information from the internet and distilling it into concise, relevant summaries.

Implementation 🛠️

Integrates a web search module using DuckDuckGo's API, a result parser, and a text summarization engine leveraging OpenAI's language models. The system performs site-specific or general searches, extracts relevant content, generates concise summaries, and compiles attributed results for efficient information retrieval and synthesis.

16. [Multi agent research team - Autogen](#)

Overview 🔍

This technique explores a multi-agent system for collaborative research using the AutoGen library. It employs agents to solve tasks collaboratively, focusing on efficient execution and quality assurance. The system enhances research by distributing tasks among specialized agents.

Implementation

Agents are configured with specific roles using the GPT-4 model, including admin, developer, planner, executor, and quality assurance. Interaction management ensures orderly communication with defined transitions. Task execution involves collaborative planning, coding, execution, and quality checking, demonstrating a scalable framework for various domains.


Additional Resources

- [comprehensive solution with UI](#)
- [Blogpost](#)

🌟 Special Advanced Technique 🌟

17. [Sophisticated Controllable Agent for Complex RAG Tasks](#)

Overview

An advanced RAG solution designed to tackle complex questions that simple semantic similarity-based retrieval cannot solve. This approach uses a sophisticated deterministic graph as the "brain"  of a highly controllable autonomous agent, capable of answering non-trivial questions from your own data.

Implementation

- Implement a multi-step process involving question anonymization, high-level planning, task breakdown, adaptive information retrieval and question answering, continuous re-planning, and rigorous answer verification to ensure grounded and accurate responses.

Getting Started

To begin exploring and building GenAI agents:

1. Clone this repository:

```
git clone https://github.com/NirDiamant/GenAI_Agents.git
```



2. Navigate to the technique you're interested in:

```
cd all_agents_tutorials/technique-name
```



3. Follow the detailed implementation guide in each technique's notebook.

Contributing

We welcome contributions from the community! If you have a new technique or improvement to suggest:

1. Fork the repository
2. Create your feature branch: `git checkout -b feature/AmazingFeature`
3. Commit your changes: `git commit -m 'Add some AmazingFeature'`
4. Push to the branch: `git push origin feature/AmazingFeature`

5. Open a pull request



Contributors



License

This project is licensed under a custom non-commercial license - see the [LICENSE](#) file for details.

Releases

No releases published

Sponsor this project



NirDiamant

Sponsor

[Learn more about GitHub Sponsors](#)

Packages

No packages published

Contributors 4

- NirDiamant
- yanivvak
- avishalom
- tucosaurus Akash Mishra

Languages

- Jupyter Notebook 100.0%

