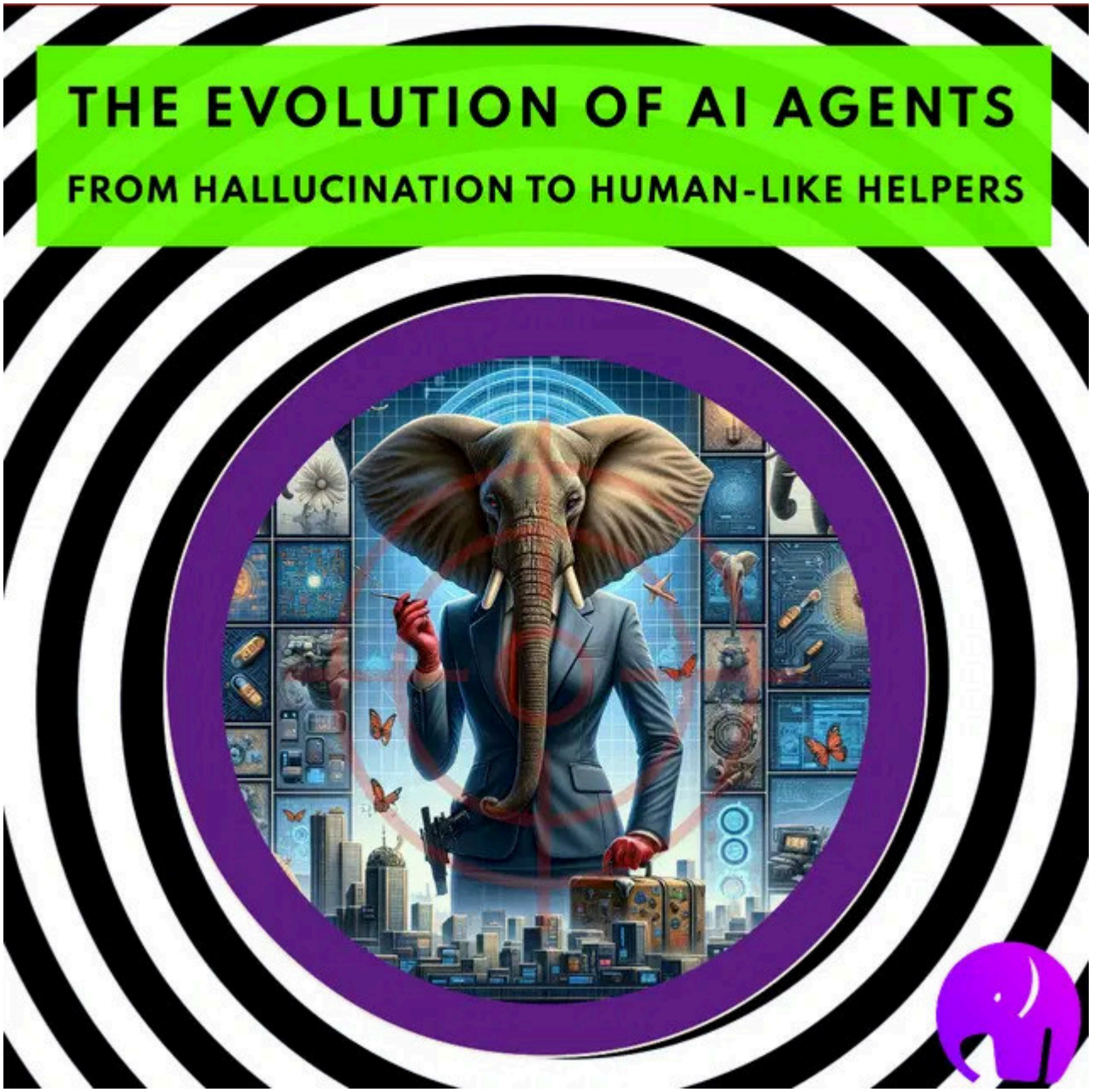
Breaking Boundaries: The Evolutionary Saga of AI Agents

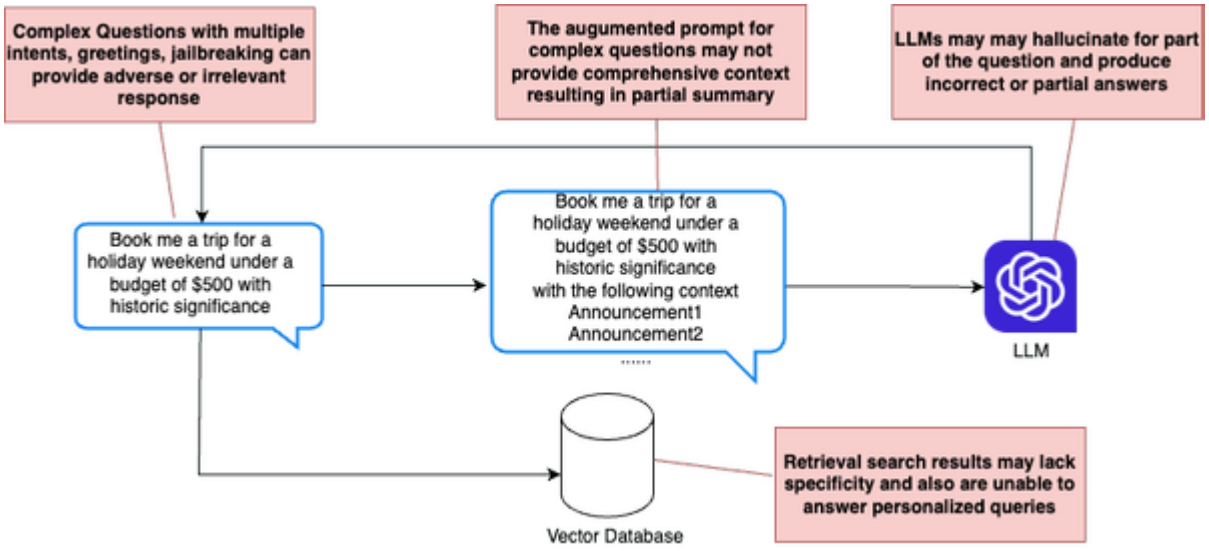


The advent of **Generative AI** has sparked a wave of enthusiasm among businesses eager to harness its potential for creating Chatbots, companions, and copilots designed to unlock insights from vast datasets. This journey often begins with the art of prompt engineering, which presents itself in various forms, including Single-shot, Few-shot, and Chain of Thought methodologies. Initially, companies tend to deploy internal chatbots to bolster employee productivity by facilitating access to critical insights. Furthermore, customer support, traditionally seen as a cost center, has become a focal point for optimization efforts, leading to the development of Retrieval Augmented Generation (RAG) systems intended to provide

deeper insights. However, challenges such as potential inaccuracies or "hallucinations" in responses generated by these RAG systems can significantly impact customer service representatives' decision-making, potentially resulting in customer dissatisfaction. A notable incident involving Air Canada has recently highlighted the potential risks to brand reputation and financial stability posed by deploying these autonomous chatbots in customer support scenarios. The prospect of creating similar chatbots for financial advisors, capable of delivering human-like yet fundamentally flawed responses, raises significant concerns. Issues related to quality (such as hallucination, truth grounding, and comprehensiveness), content safety, and the risk of intellectual property leakage are among the key hurdles preventing many generative AI applications from reaching production stages.

Challenges in achieving quality and trust

It is easy to build a simple RAG system by combining Vector search for retrieval and LLM to summarize retrieved chunks, a massive upgrade from traditional knowledge bases with a limited understanding of the semantic nature of questions. These systems show poor performance in the real world for a multipart of complex questions.



Let's deep dive into the challenges by breaking down the RAG system,

1. Question semantics:

   Complex queries often encompass multipart intents that may be unrelated or even adversarial, designed to confuse the model or "jailbreak" the chatbot. These can range from greetings to questions that test the system's limitations or probe for inconsistencies. Without understanding these nuances, a RAG system might fail to appropriately categorize and respond to the query, leading to irrelevant or incorrect answers.

2. Retrieval phase:

A single vector store search may not yield relevant results for complex or multipart statistical questions. Personalized queries, such as those asking for specific information about a user's insurance policy, pose additional challenges if the system needs access to personalized data points like the policies owned by the user. This limitation can prevent the system from providing accurate, user-specific information.

3. Prompt augmentation:

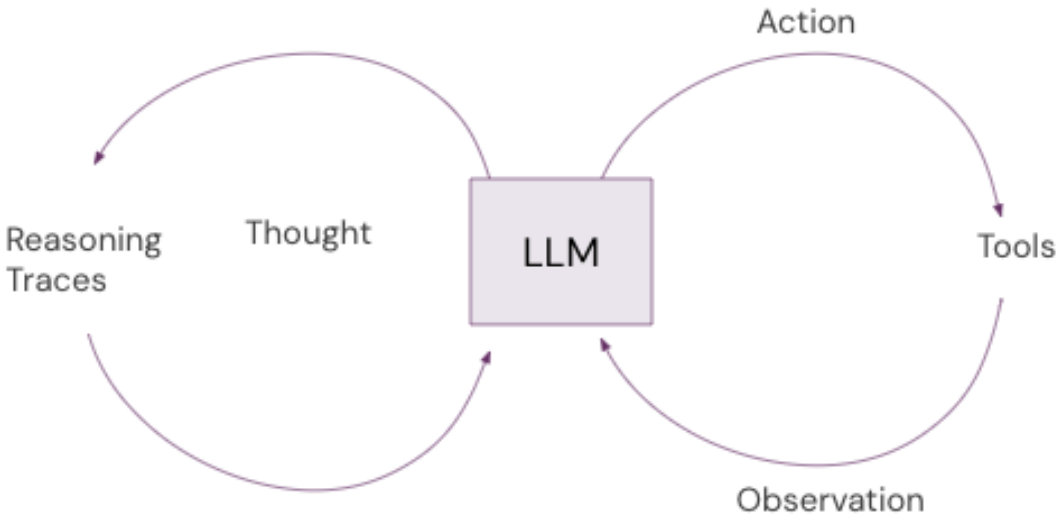
In simpler RAG implementations, the system prompt is static, combined with retrieved contextual information to create an augmented prompt. This static nature can limit the system's ability to dynamically adjust to the specifics of the query, particularly for complex or evolving scenarios that require a more nuanced understanding and response.

4. LLM for Summarization:

If the augmented prompt lacks the necessary context to answer the query effectively, LLMs may rely on their inherent knowledge base to fill in the gaps, leading to "hallucination," where the model generates plausible but inaccurate or fabricated information. This issue is particularly problematic in scenarios requiring precise, factual responses.

Rise of Agents

Prompt engineering techniques such as Chain of Thoughts (CoT) involve generating intermediate steps or reasoning paths when solving complex problems, especially in language models. It's like showing one's work in math problems but applied to AI. The model explicitly generates a sequence of thoughts or reasoning steps before arriving at a final answer or conclusion. Although CoT excels at breaking down complex tasks or questions, their effectiveness hinges on the context provided if used in RAG systems.
The ReACT (Synergizing Reasoning and Acting in Language Models) paper shows how this approach is far superior to CoTs. Let's look into the basics. In the study of autonomous agents and multi-agent systems, the concepts of Thought, Action, and Observation play crucial roles in defining how these agents perceive, interpret, and interact with their environment.



- Thought

  in AI agents refers to the internal processing or decision-making mechanisms that occur before taking an action. It involves the interpretation of observations, the weighing of possible actions based on learned experiences or predefined rules, and the formulation of a plan or response. Thought processes in AI can range from simple if-then rules to complex algorithms that involve reasoning, planning, and prediction based on deep learning models.

- Action

  is the step an AI agent takes in response to its thoughts and observations. It's the execution phase where the agent applies its decision to the environment, potentially altering its state. Actions can be physical movements, such as a robotic arm picking up an object, or digital responses, like sending a message or updating a database. The scope of actions available to an AI agent depends on its capabilities and the effectors it has to interact with its environment.

- Observation

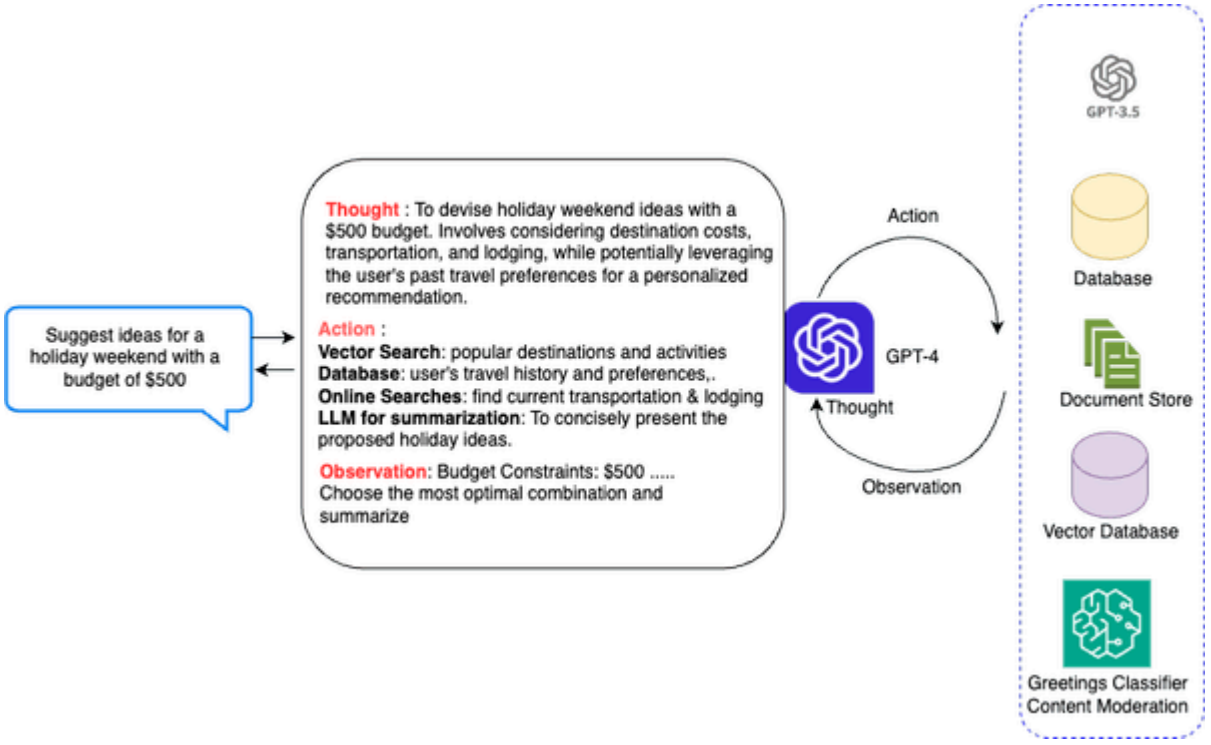
  involves the agent's perception of its environment through sensors or input mechanisms. It can include data from visual cameras, microphones, temperature sensors, or digital inputs like API calls. Observations are the raw data that an AI agent receives and processes to understand its current context or the state of the environment. Effective

observation is critical for an agent to make informed decisions and adapt actions accordingly.

Together, Thought, Action, and Observation form a cyclical process that enables AI agents to operate autonomously, learn from their environment, and achieve their goals.

RAG Agents

Agentic workflows, also known as Agents, harness the capabilities of Large Language Models (LLMs) to navigate the complexities of constructing intricate Retrieval Augmented Generation (RAG) systems. They adeptly segment elaborate tasks into manageable sub-tasks, utilize external systems to enhance their knowledge base, and monitor the outcomes to determine subsequent actions, ensuring the initial query's goals are met. The following provides a standard depiction of how a RAG system incorporates external resources for knowledge expansion.



There are several providers of Agentic solutions,

1. Langchain implements ReACT and several simple tutorials for customer service, Text 2 SQL and code interpreter.
2. LlamaIndex provides its agentic implementation using ReACT and OpenAI
3. OpenAI also introduced GPTs to create custom versions of ChatGPT by combining instructions, external knowledge, and combination of skills
4. Amazon Bedrock Agents allows you to build and configure autonomous agents in your application. An agent helps end-users complete actions based on organization data and user input. Agents orchestrate interactions between foundation models (FMs), data sources, software applications, and user conversations.

5. Semantic Kernel is an open-source project developed by Microsoft. It is an SDK that integrates Large Language Models (LLMs) like OpenAI, Azure OpenAI, and Hugging Face with conventional programming languages like C#, Python, and Java. Semantic Kernel achieves this by allowing you to define plugins that can be chained together in just a few lines of code.

Numerous options exist for creating Agentic workflows, yet they are not without challenges, including potential loops from unclear prompts or Large Language Models (LLMs) errors. Karini AI streamlines the process, enabling the rapid development and deployment of production-grade agentic workflows with the following features:

- Pre-built prompts:

  Get a head start with a comprehensive library of Agentic Prompt templates designed for various needs like customer service, HR, IT, legal, and finance. These templates save you valuable time and effort.

- Experiment and Refine:

  Seamlessly connect external tools to your workflow, enhancing your prompt creation process. Design compelling prompts and engage in interactive testing sessions with your AI agents. Analyze outcomes from top model providers and log your findings to identify best practices.

- Rapid Deployment:

  Recipes for RAGs (Retrieval Augmented Generation) expedite the deployment of your AI workflows, complete with integrated performance, usage, and cost monitoring.

- Deploy with Confidence:

  Integrate an agentic co-pilot directly into your systems. Choose from optional safety features for added peace of mind.

- Recipes for RAGs:

  expedite the deployment of agentic workflows, complete with integrated performance, usage, and cost monitoring.Create custom greetings to enhance user experience. Continuously improve your AI with a built-in feedback mechanism.

Karini AI empowers you to build, deploy, and manage powerful AI agents efficiently. Start your journey today!

Conclusion:

The ReAct agent represents an advanced form of artificial intelligence, drawing inspiration from the human processes of thinking, acting, and observing to tackle challenges methodically. Whether you're a Generative AI aficionado or looking to gain a competitive edge by creating production-level agents through an intuitive visual platform, the Karini AI platform is designed to accelerate your journey to market with ethical AI solutions.

About Karini AI:

Fueled by innovation, we're making the dream of robust Generative AI systems a reality. No longer confined to specialists, Karini.ai empowers non-experts to participate actively in building/testing/deploying Generative AI applications. As the world's first GenAIOps platform, we've democratized GenAI, empowering people to bring their ideas to life – all in one evolutionary platform.

Contact Us:

Jerome Mendell
(404) 891-0255
sales@karini.ai
https://www.karini.ai/