Kaggle | Google

5-Day Gen AI Intensive Course with Google 2025

Whitepaper Companion Podcast (Notes): Agents Companion

AI Agents White Paper Companion: A Deep Dive

Introduction to Generative AI Agents

- Generative AI agents represent a significant advancement from traditional language models.
- Agents are designed to achieve specific objectives by:
    - Perceiving their environment.
    - Strategically acting upon it.
    - Utilizing available tools.
- The core of an agent involves integrating reasoning, logic, and external information access.
- Agents can operate autonomously, pursuing goals and determining actions without explicit instructions.

Advanced Agent Concepts for Developers

- This white paper aims to provide developers with a shortcut to understanding advanced agent concepts and emerging best practices.
- It acknowledges the shift from proof of concept to building reliable agents in live environments.

Agent Ops: Tailored DevOps for AI Agents

- Agent Ops is a hybrid of DevOps and MLOps, customized for managing AI agents.

- Key components include:
    - Robust processes for managing agent tools.
    - Orchestrating complex workflows.
    - Efficient memory handling.
    - Breaking down large tasks into smaller, manageable units.
- The goal is to treat agents as a well-oiled machine, with each part functioning smoothly and a system in place for monitoring and maintenance.

Measuring Success with Business KPIs

- Business KPIs, such as goal completion, user engagement, and revenue, should be central in measuring agent success.
- Agents are tools designed to achieve specific business goals, so their success should be measured accordingly.
- Instrumentation is needed to capture granular metrics that feed into these KPIs.
    - Task success rate.
    - User interaction.
    - Detailed logs of agent actions for debugging.
- Tracking provides insights into both the outcome and the path the agent took to get there.

The Importance of Human Feedback

- Human feedback is invaluable, gathered through:
    - Thumbs up/down systems.
    - User surveys.
    - Open-ended feedback.
- Real-world insights into how the agent is perceived and performs are crucial.
- User experience feedback is valuable and necessary for a complete picture.

Automated Evaluation

- Automated evaluation is essential for assessing agent capabilities, analyzing problem-solving trajectories, and judging the quality of final responses.
- It automates the quality control process for agents.

Analyzing Agent Trajectory

- Evaluating an agent's trajectory involves analyzing the path taken to reach a solution.
- Key considerations:
    - Did the agent pick the right tools?
    - Was the approach efficient?
    - Did it waste time exploring dead ends?

Techniques for Measuring Trajectory

- **Exact Match:** Compares agent actions to a predefined ideal sequence.
- **In-Order Match:** Provides flexibility as long as core steps are followed in the correct order.
- **Any-Order Match:** Used when the sequence itself is not critical.

Judging the Quality of Final Response

- Auditors, using one LLM to judge the output of another LLM, can be employed.
- Criteria for a good response are set, and the auditor compares the agent's output to those standards.
- This functions as an AI quality assurance team working continuously.

The Role of Humans in Evaluation

- Humans remain a vital part of the evaluation process.
- Auditors are effective for objective assessments, but humans are better at judging:
    - Creativity.

    - Common sense.
    - Nuance.
- Human feedback ensures automated evaluation methods align with user needs and expectations.

Multi-Agent Systems: Divide and Conquer

- Instead of one agent handling everything, complex problems are broken down into smaller tasks.
- Specialized agents are assigned to handle each task.
- This approach leverages specialization, much like a real-world team.

Benefits of Multi-Agent Systems

- **Accuracy:** Agents can double-check each other's work.
- **Efficiency:** Agents can work in parallel, speeding up processes.
- **Scalability:** Processing power can be increased by adding more agents.
- **Fault Tolerance:** If one agent fails, others can take over.
- **Mitigation of AI Issues:** Combining perspectives reduces the impact of biases or hallucinations.

Structuring Multi-Agent Systems: Design Patterns

- Design patterns are blueprints for building and organizing multi-agent teams.
    - Sequential Pattern
    - Hierarchical Pattern
    - Collaborative Pattern
    - Competitive Pattern

Sequential Pattern

- Agents work in a chain, each completing a task and passing the output to the next.

Hierarchical Pattern

- A manager agent oversees a team of worker agents, delegating tasks and coordinating the workflow.

Collaborative Pattern

- Agents work as equals, sharing information and resources to achieve a common goal.

Competitive Pattern

- Agents compete against each other to find the best solution, effective for optimization problems.

Challenges in Building Multi-Agent Systems

- Task Allocation
- Coordination of Reasoning Processes
- Managing Context Volume
- Time and Cost

Evaluating Multi-Agent Systems

- Techniques like trajectory analysis and final response evaluation can be applied.
- Need to assess how individual agents perform and how well they coordinate.
- Questions to ask:
    - How effectively are agents cooperating?
    - Are they communicating clearly?
    - Are the right agents assigned to the right tasks?

Agentic RAG: Retrieval Augmented Generation with Agents

- Agentic RAG incorporates intelligent agents into the retrieval process.
- Agents refine search queries, evaluate retrieved information, and adapt to new knowledge.
- This enhances accuracy, contextual understanding, and adaptability.

Optimizing Basic Search Engine

- Ensure effective parsing and chunking of source documents.
- Add relevant metadata (synonyms, keywords, authors, dates, tags, categories) to chunks.
- Fine-tune the embedding model or use a search adapter.
- Use a faster vector database.
- Implement rankers to re-rank results and ensure the most relevant ones appear at the top.

Google Cloud Tools for Search

- **Vertex AI Search:** A powerful search engine offering Google-quality search for your data.
- **Vertex AI Search Builder APIs:** Provide flexibility to create custom search engines.
- **Vertex AI RAG Engine:** Orchestrates the entire RAG pipeline.

Real-World Example: Google's Co-scientist

- Co-scientist is a multi-agent system designed to accelerate scientific discovery.
- It uses specialized agents to generate, evaluate, and refine hypotheses.
- In one study, it identified existing drugs for liver fibrosis and proposed new mechanisms and drug candidates.

Automotive AI: Multi-Agent Systems in Cars

- Modern cars require systems for navigation, media management, answering questions, and handling general knowledge queries.

- Specialized agents are emerging for each of these tasks.
    - Conversational navigation agent.
    - Media search agent.
    - Car manual agent.
    - General knowledge agent.

Patterns in Automotive AI

- **Hierarchical:** A central orchestrator agent routes queries to specialists.
- **Diamond:** Responses are filtered through a central moderator agent for tone and style adjustments.
- **Peer-to-Peer:** Agents communicate directly with each other, creating a decentralized system.
- **Collaborative:** Multiple agents work together to answer complex questions.

Benefits of Multi-Agent Approach in Automotive AI

- Specialization leads to higher quality responses.
- Efficiency and speed are increased by matching the right resources to each task.
- Resilience ensures the system continues operating even if one agent encounters an issue.

Agent Builder: Google Cloud's Toolkit for Agent Developers

- Agent Builder is a suite of products and services designed for agent developers.
- It includes:
    - **Vertex AI Agent Engine:** Simplifies agent development and deployment.
    - **Vertex AI Evaluation Service:** Offers evaluation tools for LLMs, RAG systems, and agents.

Agents as Contractors: A Conceptual Shift

- As AI agents become more autonomous, giving them simple instructions may not be enough.
- The paper suggests applying principles from real-world contracts to AI, setting expectations, managing risk, and resolving disputes.
- This raises questions about what an AI contract would look like and how it would be enforced.
- The goal is greater accountability and transparency as AI becomes more powerful.

Agentic RAG Deep Dive

- Agentic RAG adds intelligent agents to the traditional RAG process.
- Agents refine search queries, evaluate retrieved information, and adapt to new knowledge.
- Example: A car manual agent breaks down user questions into smaller parts and formulates specific search queries for each step.

Optimizing Search for Agentic RAG

- Effectively parse and chunk source documents.
- Add relevant metadata.
- Fine-tune the embedding model or use a search adapter.
- Use a faster vector database.
- Implement rankers.
- Incorporate check grounding to ensure claims are supported by retrieved information.

Conceptual Shifts: Trust and Reliability

- As AI agents play a larger role, trust and reliability become essential.
- Factors contributing to trust and reliability:

- Transparency
- Accountability
- Robustness
- Fairness

Transparency

- The more we understand how an agent works, the more likely we are to trust its decisions.
- Transparency can involve:
  - Providing clear explanations of how the agent arrived at a conclusion.
  - Giving users the ability to audit the agent's actions.

Accountability

- Accountability ensures someone or something is responsible for the consequences of an agent's decisions.
- This can be achieved through:
  - Monitoring and auditing agent behavior.
  - Developing legal frameworks that hold AI developers and operators responsible.

Robustness

- Robustness is the ability of an agent to handle unexpected situations or inputs without crashing or behaving erratically.
- This can be built into agents by:
  - Testing them in a wide range of scenarios.
  - Incorporating failsafe mechanisms.
  - Designing them with the ability to learn and adapt.

Fairness

- Fairness ensures that AI agents do not discriminate against certain groups or perpetuate existing biases.
- This starts with:
    - The data used to train them.
    - The algorithms used to make decisions.
    - The metrics used to evaluate their performance.