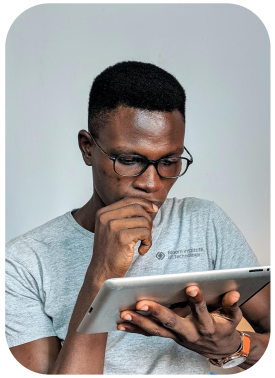
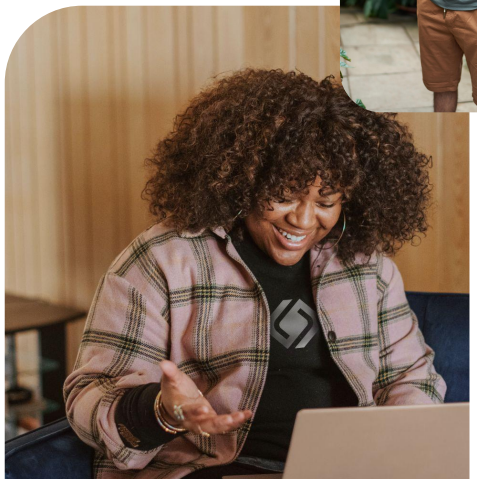
Bloom Institute of Technology

Advanced Graph Techniques

Designing and launching human-in-the-loop systems, complex agents within graphs, corrective/adaptive RAG, and ranking outputs using DyLAN.





Core Competencies

The student must demonstrate...

1. When should you use graphs within agents workflows (5 min)
2. Understanding Human-in-the-loop design patterns (10 min)
3. Evaluating and ranking agentic outputs with DyLAN (10 min)
4. Adaptive and Corrective RAG Techniques (10 min)
5. Previewing Final Project (5 min)

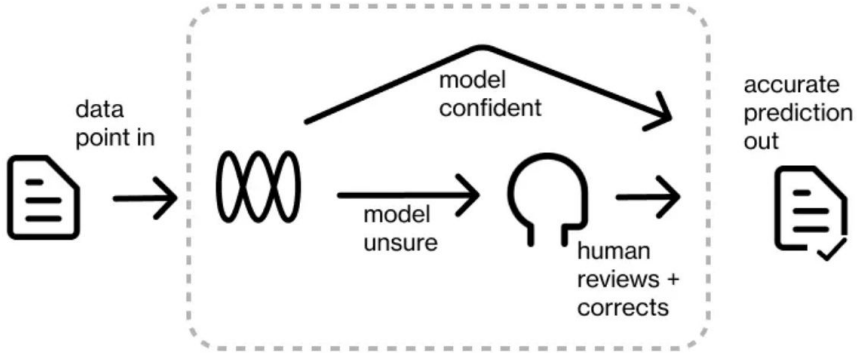
Single-Agent vs. Multiple Agents Using Graphs

	Single-Agent Patterns	Multi-Agent Architectures
Best Suited For	Tasks with a narrowly defined list of tools and well-defined processes	Tasks benefiting from feedback from multiple personas and division of tasks
Implementation	Easier, only one agent and set of tools to define	More complex conversation management and clear leadership
Risks	Poor feedback and potential execution loops if reasoning isn't robust	Requires more management to avoid distractions and reach goal
Advantages	Simpler setup, avoids issues like poor feedback from other agents	Better performance in parallel tasks or workflows, handles feedback well
Use Cases	Ideal when the path is unclear, requires thoughtful next steps	Useful for tasks needing parallelization and diverse feedback

Key Takeaway: Use multiple agents within graphs for tasks benefiting from feedback and parallelization. This enhances performance in complex scenarios requiring multiple perspectives and specialized functions.

Human-in-the-loop Design Patterns

Initially, use humans to train and refine the agent's output through iterative feedback. As the agent's performance improves, gradually reduce human intervention until it operates autonomously.



Accessed 4 June 2024, <https://humanloop.com/blog/human-in-the-loop-ai>.

Impact of Human Oversight on Agent Systems

- Iterative feedback is crucial for solving complex problems.
- Language models often commit early, missing the goal entirely.
- Feedback helps agents correct their course.
- Agents can be misled by feedback from other agents, leading to faulty plans.
- Robust prompting mitigates risks; humans should have the final say.

Examples of Human Oversight

1. Retail AI: Human reviews AI-written Google review responses before posting.
2. Analytics AI: Data engineer checks AI-extrapolated purchase history before database insertion.
3. Travel AI: Human confirms trip reservations (flights, hotels, etc.) before purchase.

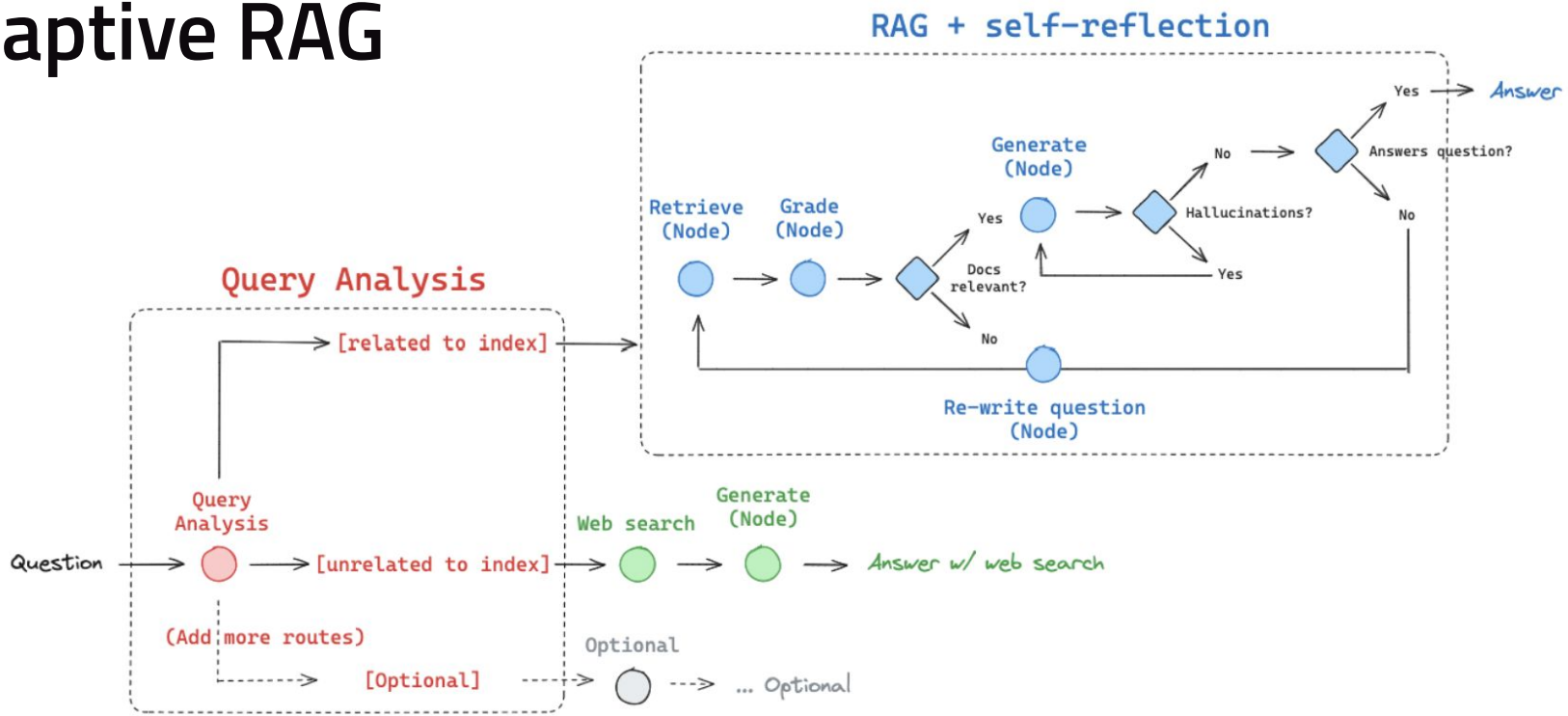
Dynamic LLM-Agent Network (DyLAN)

Access the official DyLAN repo by SALT-NLP.

How DyLAN Works: Step-by-Step

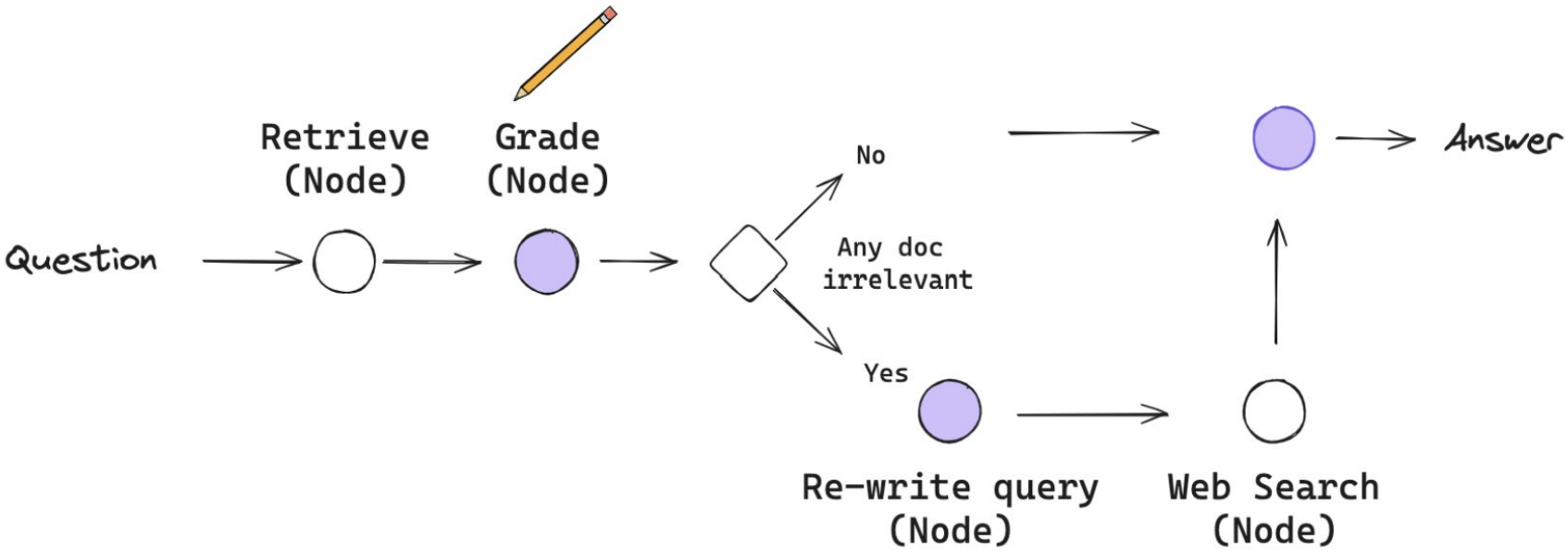
1. **Form Team**: DyLAN assembles a group of agents to tackle complex tasks like code generation.
2. **Task Execution**: Agents collaborate, each contributing their expertise.
3. **Evaluate Performance**: After each round, DyLAN assesses how well each agent performed.
4. **Select Top Performers**: The best-performing agents continue to the next round.
5. **Share Information**: Agents share their knowledge without a designated leader.
6. **Repeat and Refine**: The process repeats, with DyLAN continuously selecting top agents.
7. **Achieve Better Results**: This iterative approach leads to improved outcomes in general reasoning tasks.

Adaptive RAG



Accessed 4 June 2024, <https://github.com/langchain-ai/langgraph/tree/644ea3a3d7aa89833b0f073444fbca709a9a7b98/examples/rag>.

Corrective RAG



Accessed 4 June 2024, <https://github.com/langchain-ai/langgraph/tree/644ea3a3d7aa89833b0f073444fbca709a9a7b98/examples/rag>.

Final Project Preview

In the last two weeks of the course, you will **build a final project designed for direct implementation within your team at your company**. Our goal is to provide you with tangible code and guidance on creating practical solutions, not just theoretical knowledge about using AI agents.

Overview of Tasks

1. **Identify a Problem**: Work by yourself or collaborate with your team to find a problem suitable for automation or AI solutions.

2. **Design the Solution**: Plan the implementation, including the agent's tasks, training strategy, and performance metrics.

3. **Code a Prototype**: Develop a functional prototype, integrating the agent with necessary systems. The goal is MVP, not something with every single feature you want.

4. **Test in Your Ecosystem**: Deploy and test the prototype, gathering feedback and refining it based on real-world performance.

Hands-On Homework.

Start brainstorming your final project goals:

1. **Identify a Problem:** Collaborate to find a problem suitable for automation or AI solutions.

2. **Design the Solution:** Plan the implementation, specifying the agent's tasks, training strategy, and performance metrics.