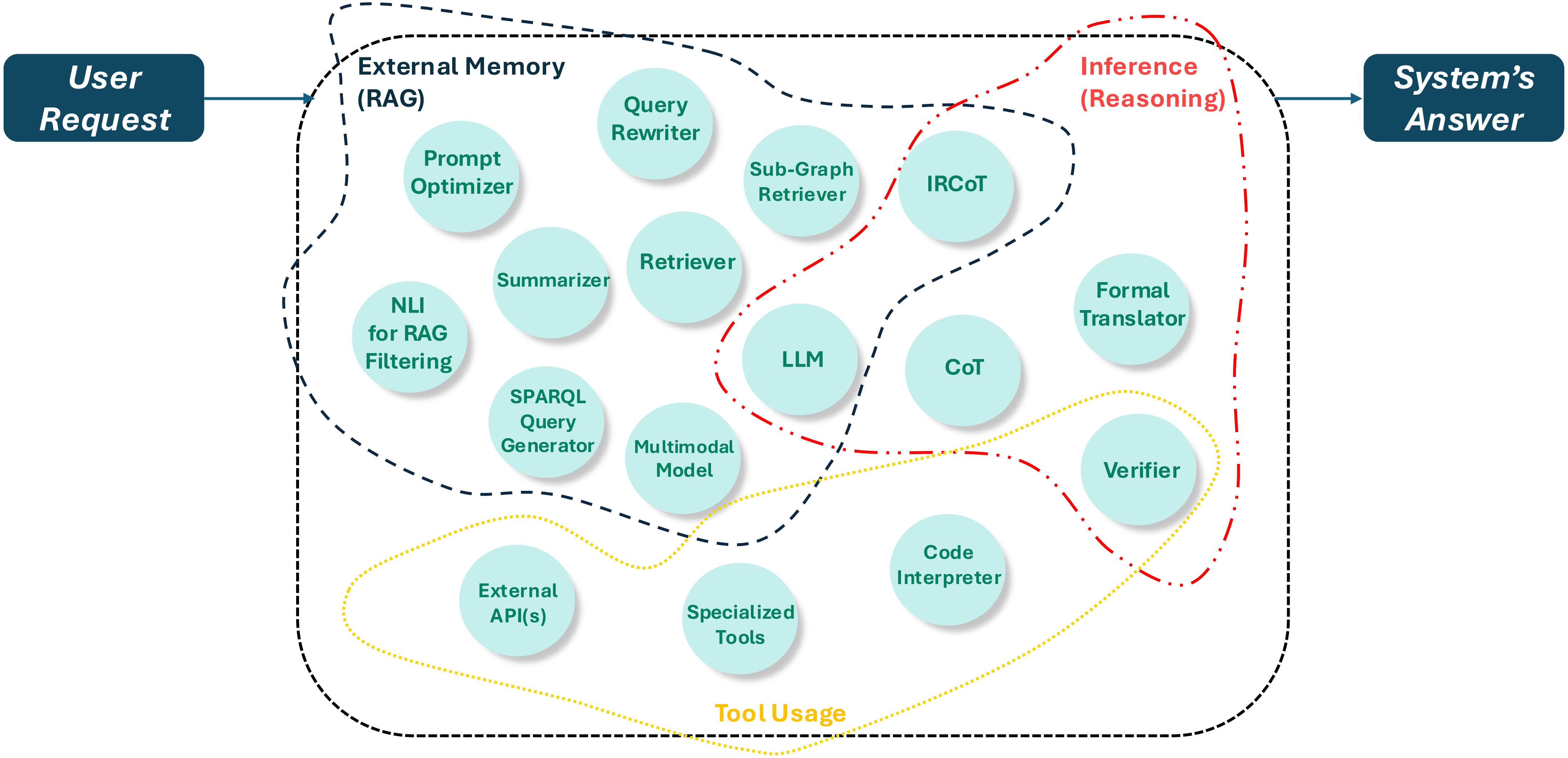
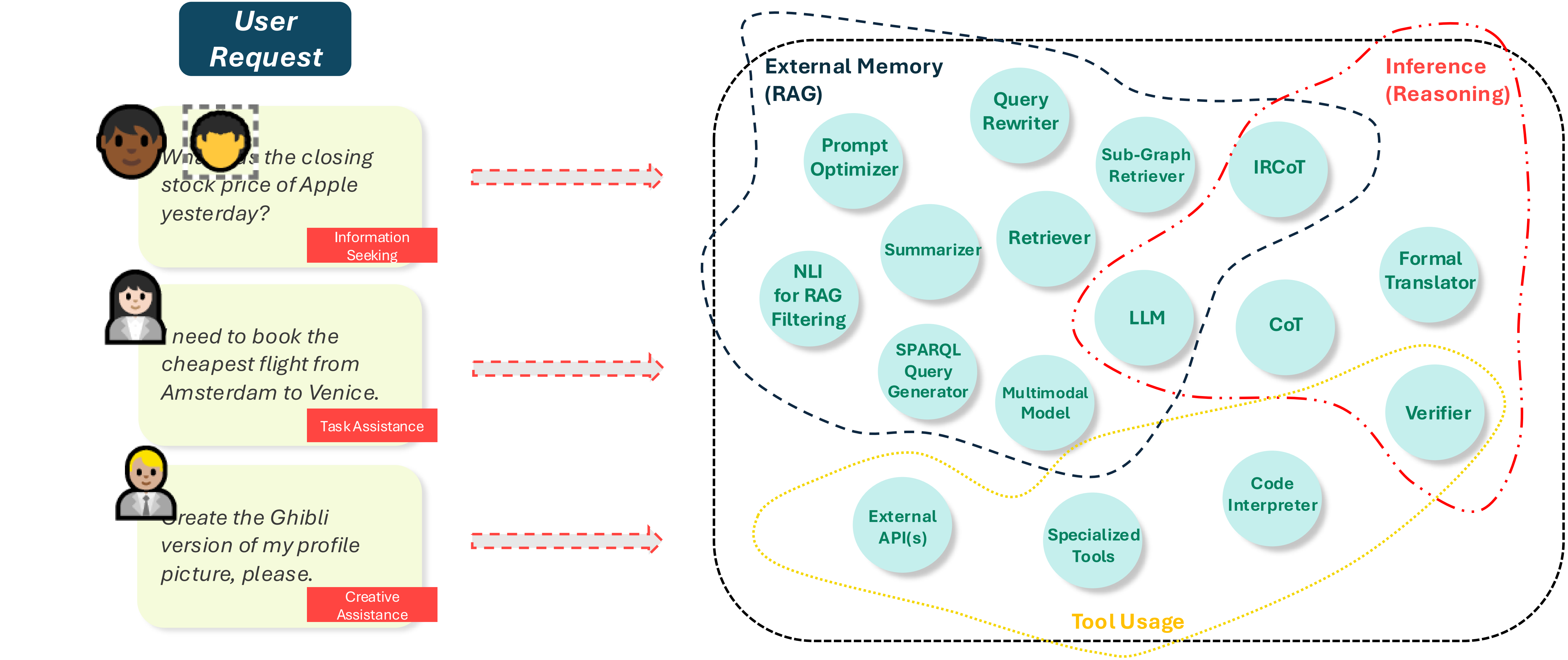
Adaptive Orchestration of Modular Generative Information Access Systems

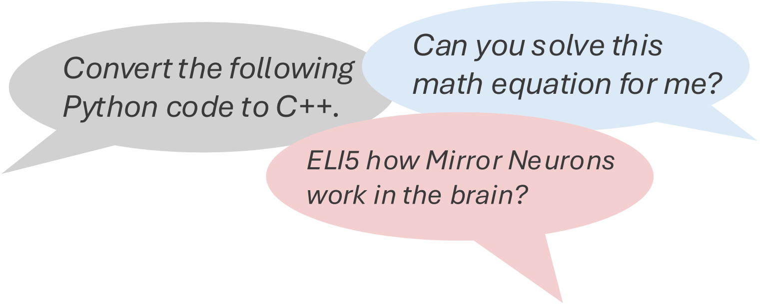
Mohanna Hoveyda, Harrie Oosterhuis, Arjen P. de Vries, Maarten de Rijke, Faegheh Hasibi

SIGIR 2025
Padova
ITALY



Generative Information Access System
(GenIA)





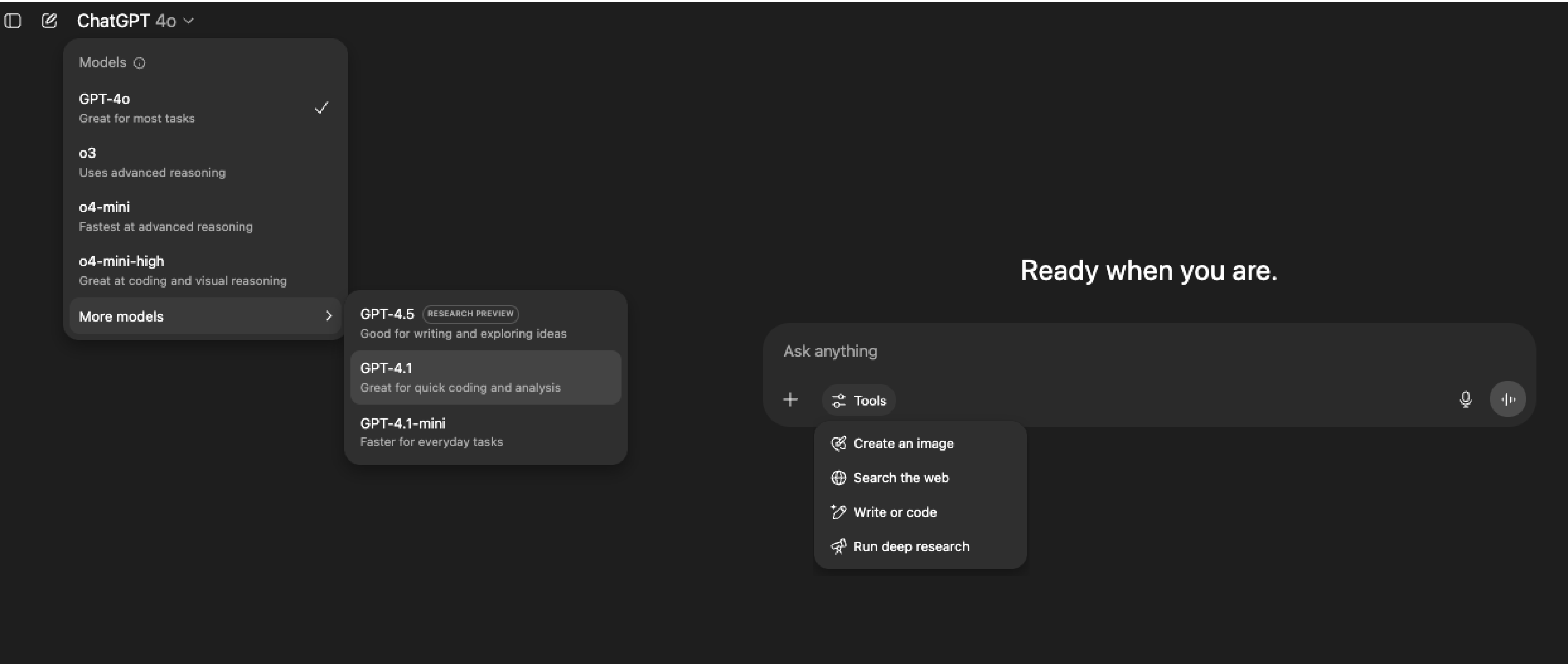
Generative Information Access System

(GenIA)

Perspective

Designs of future GenIA Systems
should be
Dynamic, **Modular** and **Adaptive**

Where we stand now:



Q1

How to characterize the **space of possible modules** and their **interactions**?

Q2

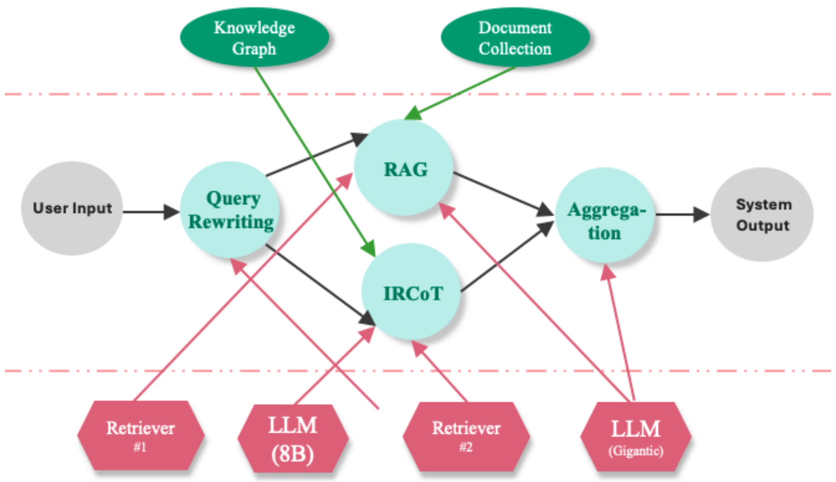
How can the system **adapt** to **diverse user queries** and **evolving modules**?

Q3

How to **automate** and **optimize interactions** between **diverse components**?

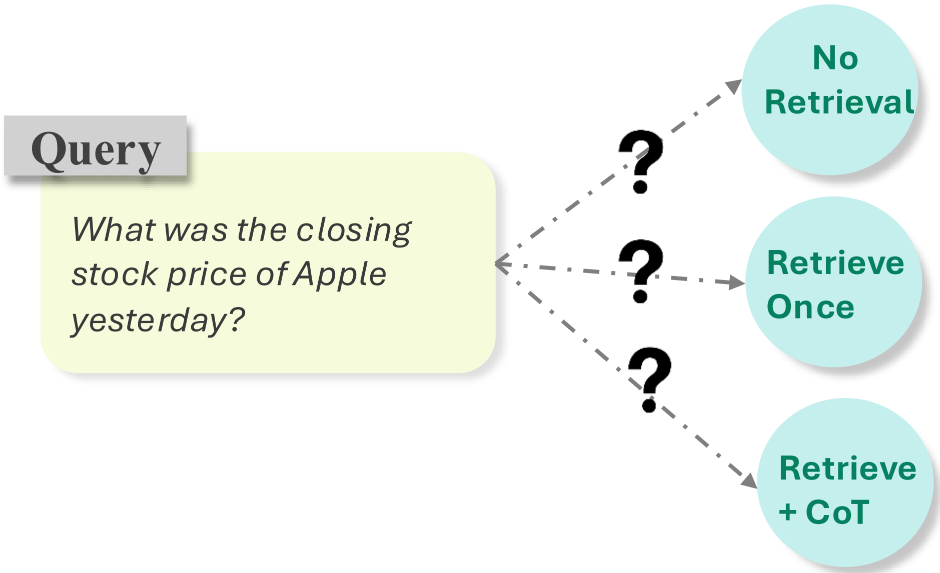
Framework

A graph-based framework for adaptive orchestration via Reinforcement Learning

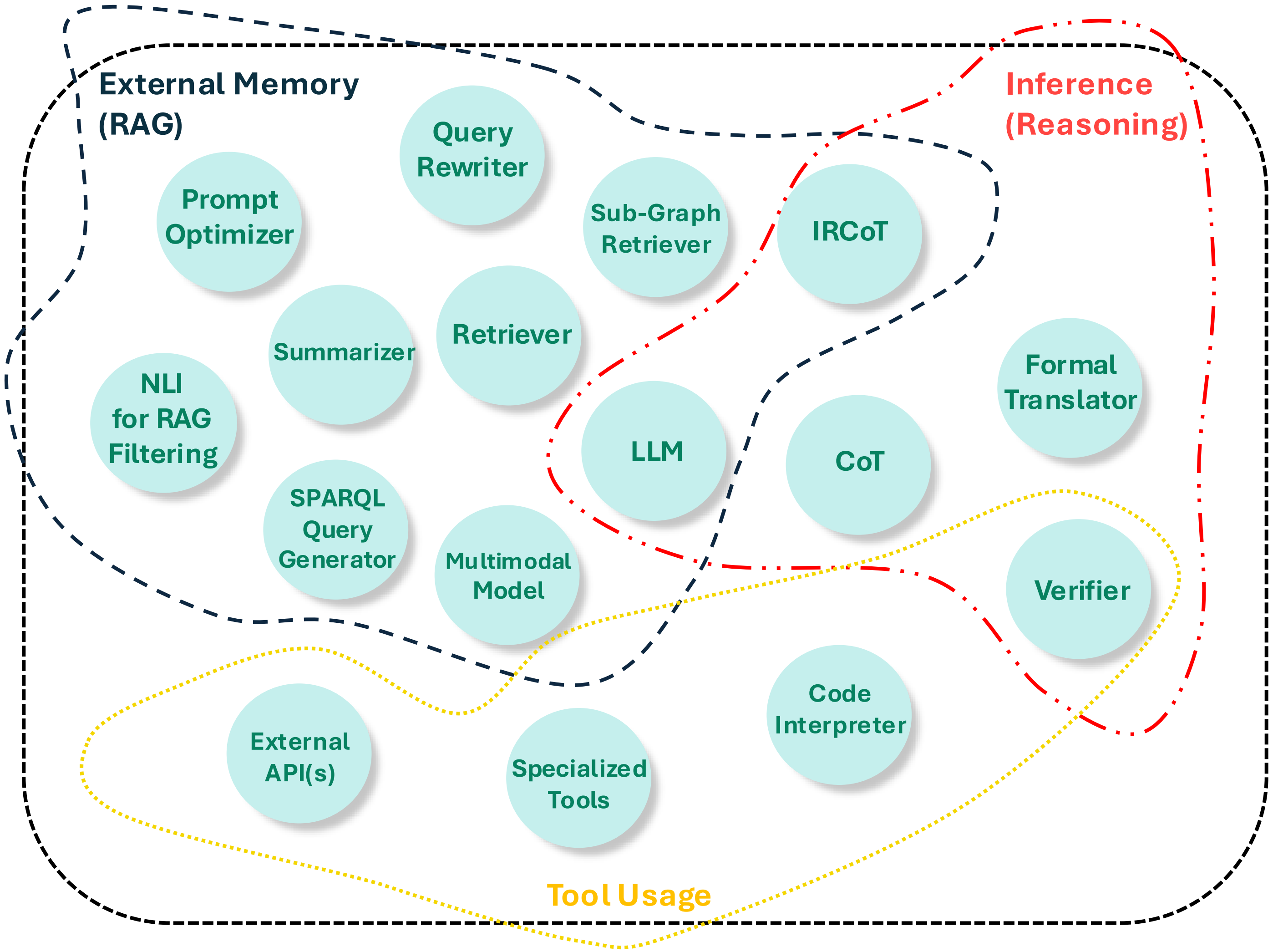


Instantiation

Adaptive Question Answering (AQA)



Q1 How to characterize the **space of possible modules** and their **interactions**?



Q1 How to characterize the **space of possible modules** and their **interactions**?

Resources

- Knowledge Graph
- Multimodal Corpus
- Proprietary Collection
- Private Collection
- Document Collection

Tasks

Standalone

- Generation
- Retrieval
- Query Rewriting
- Asking Clarifying Question
- Formal Translation
- Action Execution
- Aggrega-tion
- Verifica-tion
- Intent Recognition
- CoT-Decoding
- Recomm-endation
- Entity Linking

Complex

- RAG
- LINC
- Self-RAG
- IRCoT

Executors

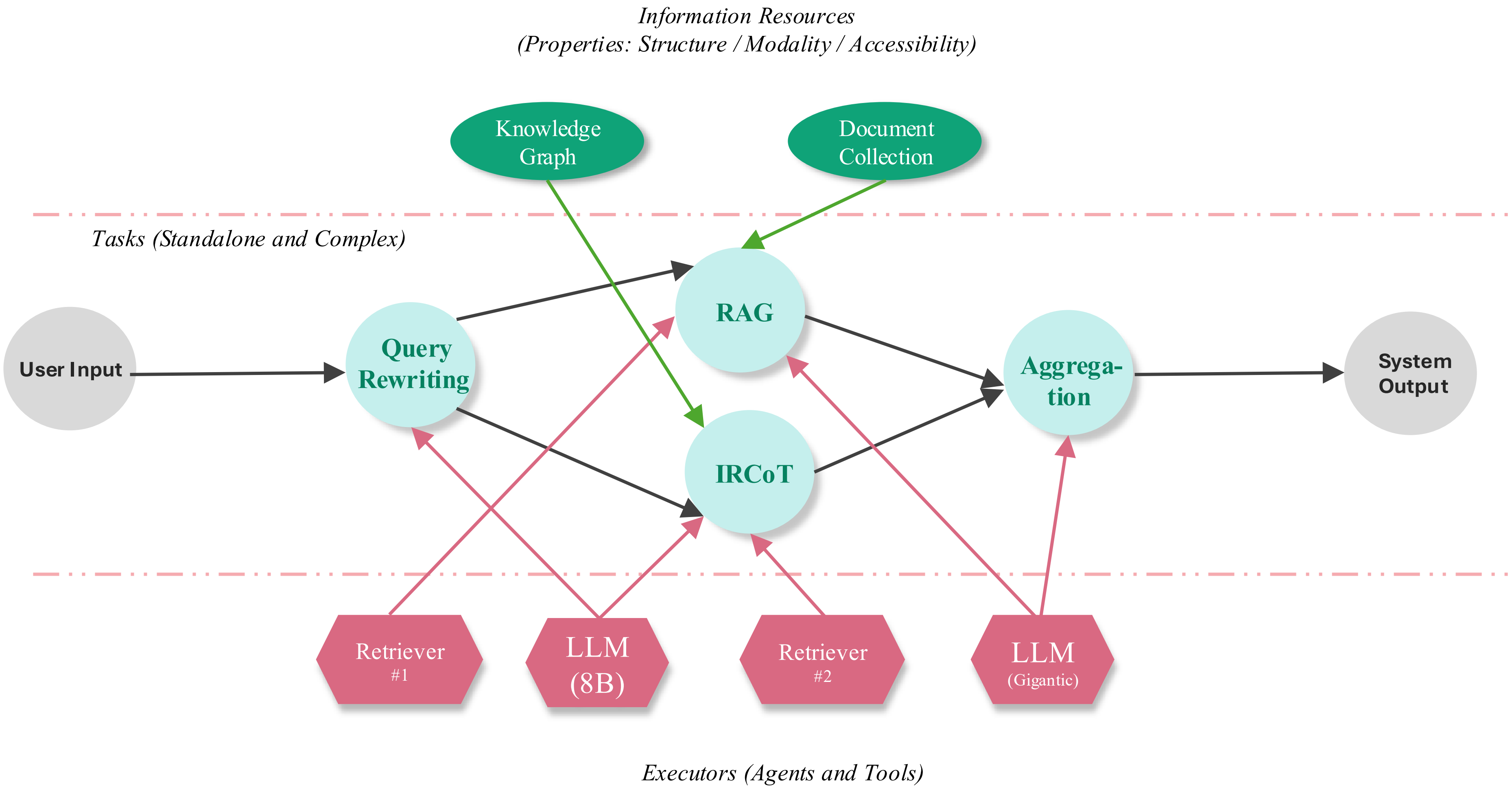
Agents

- VLMs
- LLM (8B)
- LLM (70B)
- LLM (Gigantic)
- Speciali zed LLM
- Speech Model

Tools

- Retriever #1
- Retriever #2
- External APIs
- Verifier/ Solver

Q1 How to characterize the **space of possible modules** and their **interactions**?



Graph-based representation of pipelines

Q2 How can the system **adapt** to **diverse user queries** and **evolving modules**?

#1 Representation (embedding) of the user input

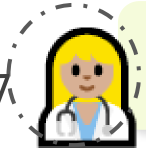
Encoding **semantics**, **syntactic**, **complexity**, **thematic and topical** info.

#2 User Profile Features

Preferences, historical **patterns**, typical complexity, frequently used resources, location, …

#3 Implicit Behavior Signals

Interaction patterns, facial expressions, eye movements

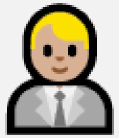


User input's text

Query context features

#1 User behavior and feedback

Implicit/Explicit indications of satisfaction



#2 Supervised Examples

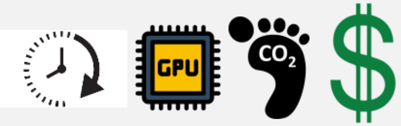
Ground-truth labels

#3 Self-supervised Objectives

Distillation or Confidence scores/UE elicited from the agents

#4 Cost Considerations

e.g. Latency; how long the user awaits an answer



Objectives of the pipeline construction

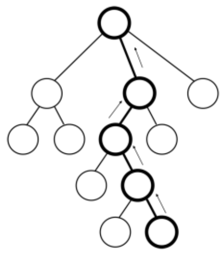
Q3 How to **automate** and **optimize interactions** between **diverse components**?

a) Constructing pipeline graphs on-the-fly based on contextual features, and
b) Learning what pipelines work best for which queries

Classical Planning Algorithms

- Scenarios where effectiveness and efficiency of pipelines can be accurately estimated.
- Pipeline construction = Search over possible graphs

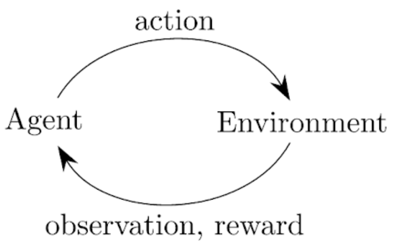
A*, beam search, Monte Carlo tree search



General Reinforcement Learning

- Pipeline construction is formulated as an MDP
- State = Partial pipeline + Contextual features
- Action = Adding a node/edge
- Reward = Pre-defined objectives

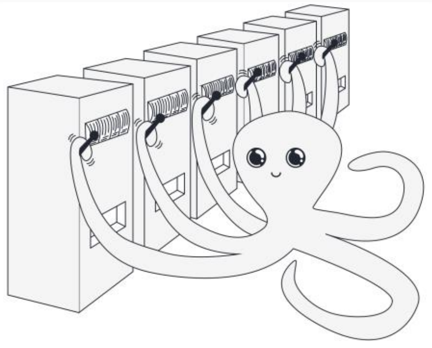
REINFORCE, Q-Learning, Actor-Critic



Contextual Bandit Algorithms

- Scenarios with limited number of possible graphs, and immediate independent rewards

LinUCB, Thompson Sampling



Methodologies for pipeline construction

Instantiation

Adaptive Question Answering
via Contextual Multi-Armed Bandits
(AQA)

Training LinUCB

$$x_t = [x_{t1}, x_{t2}, \ldots, x_{td}]$$

Question Features (Context)


```
array([-0.5968882 , -0.33086956, -0.32643065, -0.3670732 ,  0.628059  ,
       -0.3692328 , -0.37902787, -0.12308089, -0.38124698, -0.03940517,
        0.2260839 ,  0.10852845, -0.2873811 , -0.42781743,  0.06604357,
       -0.07114276, -0.29775023, -0.99628943, -0.54497653, -0.11718027,
       -0.15935768,  0.09587188, -0.2503798 ,  0.06768776,  0.3311586 ,
        0.43098116,  0.06936899,  0.24311952,  0.14515282,  0.19245838,
        0.10462623, -0.45676082,  0.5662387 ,  0.69908774,  0.48064467,
        0.27378514, -0.45430255,  0.17282294, -0.40275463, -0.38083532,
        0.47487524,  0.31950948, -0.1109335 ,  0.2165357 ,  0.034114  ,
        0.05689918,  0.20939653,  0.15209009, -0.24204595,  0.03478364,
        0.1616051 , -0.5827333 , -0.47017908,  0.26226178, -0.11884775,
        0.40180743, -0.5173988 , -0.19270805,  0.660391  , -0.24518126,
       -0.42860952, -0.22274768,  0.4887834 ,  0.49302152,  0.38799986,
       -0.041193  , -0.38600504, -0.37632987,  0.04570564,  0.50462466,
       -0.14396502,  0.33490512, -0.15964787, -0.21363072, -0.25445372,
        0.52389127,  0.5747422 , -0.25075617, -0.5339069 ,  0.2582965 ,
       -0.16139959,  0.09748188,  0.04540966, -0.27768216, -0.51260656,
       -0.06189002, -0.54032195, -0.21863565,  0.06233869,  0.13287479,
        0.49741864,  0.1772418 ,  0.02064824, -0.04775626, -0.16804916,
        0.4643644 ,  0.5546319 ,  0.68051434,  0.7790246 ,  0.5617202 ],
      dtype=float32)
```


1. Compute (for all possible graphs):

$$UCB \leftarrow Estimated\ Reward + (Exploration\ Parameter\ \times\ Uncertainty)$$

How good the graph has been in similar context before

Have we tried the graph enough before to be sure of its performance?

2. Choose & Execute

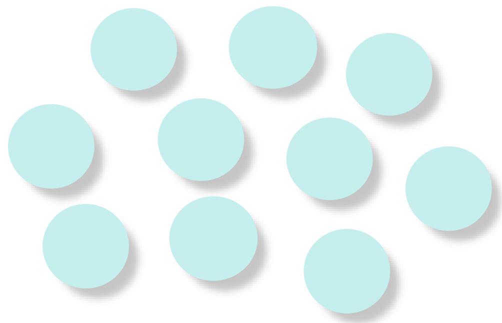
The graph with the highest UCB

3. Observe Reward (& update parameters)

$$r_t \leftarrow \beta \cdot P_t - \gamma \cdot T_t$$

Performance of the executed graph (F1-Score)

Execution Time cost of the graph



All Possible Graph Configs (Action Space)

Final Goal

Upon Training, Parameters Converge so that:

The Upper Bounds of the Best Answering Strategies Are Higher

Training LinUCB

$$x_t = \begin{cases} [1,0,0] & \text{if easy} \\ [0,1,0] & \text{if moderate} \\ [0,0,1] & \text{if complex} \end{cases}$$

Limited Features

1. Compute (for all possible graphs):

$$UCB \leftarrow Estimated\ Reward + (Exploration\ Parameter\ \times\ Uncertainty)$$

How good the graph has been in similar context before

Have we tried the graph enough before to be sure of its performance?

NoR

OneR

Final Decision

IRCoT

Limited Graphs

2. Choose & Execute

The graph with the highest UCB

Final Goal

Upon Training, Parameters Converge so that:

The Upper Bounds of the Best Answering Strategies Are Higher

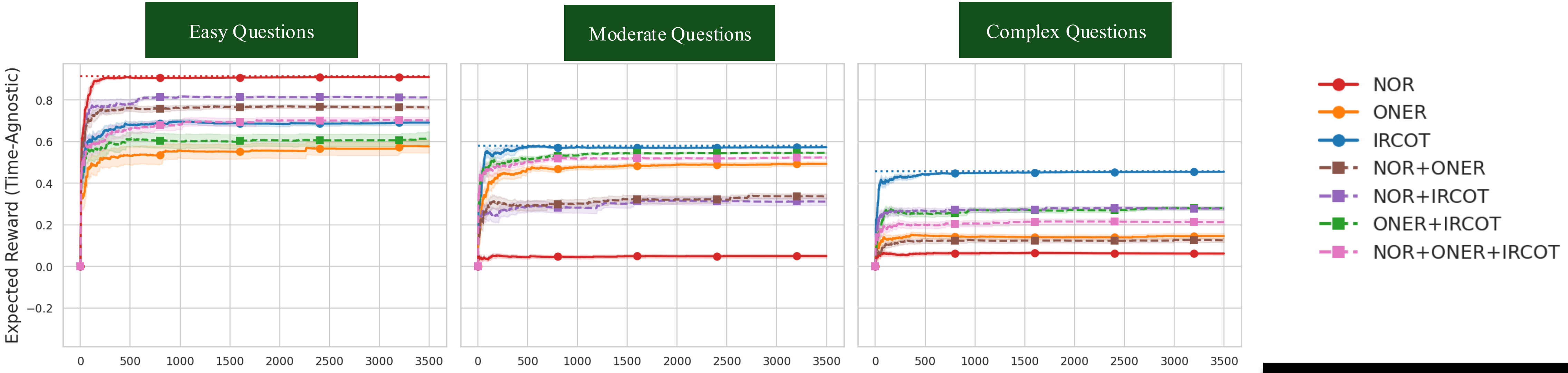
3. Observe Reward (& update parameters)

$$r_t \leftarrow \beta \cdot P_t - \gamma \cdot T_t$$

Performance of the executed graph (F1-Score)

Execution Time cost of the graph

$r_t \leftarrow \beta \cdot P_t$



Final Goal

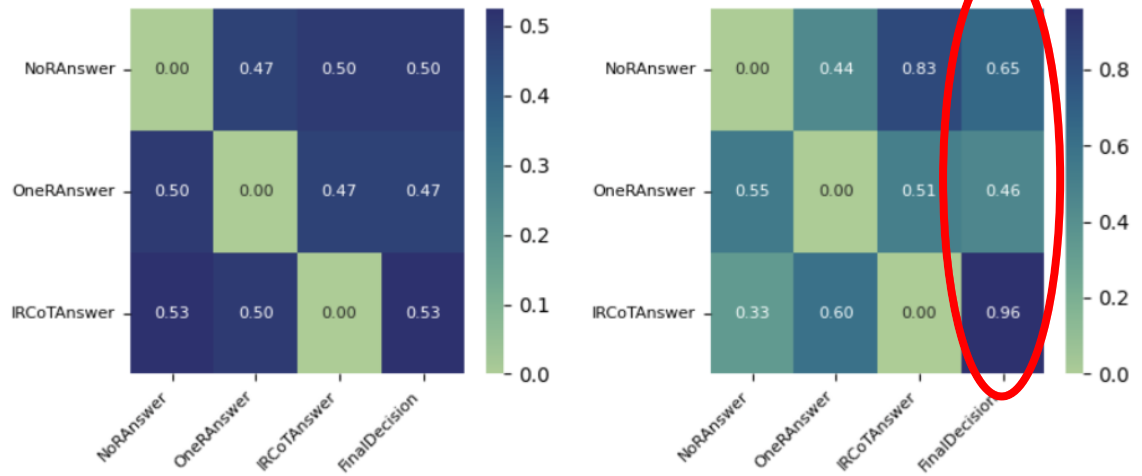
Upon Training, Parameters Converge so that:

The Upper Bounds of the Best Answering Strategies Are Higher

LinUCB expected rewards. Dashed line depicts real reward for the optimal action.

AQA vs GPTSwarm

Converged towards higher probability for the most sophisticated Agent



Edge probabilities distribution among NoR, OneR, IRCoT, and FinalDecision nodes before (left) and after (right) optimization using GPTSwarm.

1. AQA outperforms GPTSwarm in overall score
2. AQA learns to balance accuracy and time efficiency

	AQA (NT)		AQA (T)		GPTSwarm	
	F1	Time	F1	Time	F1	Time
Context A	1.0	6.18	1.0	6.18	0.862	12.78
Context B	0.568	12.04	0.539	8.73	0.327	12.79
Context C	0.523	11.75	0.523	11.75	0.317	12.76
Overall	0.697	9.99	0.687	8.89	0.502	12.78

AQA (NT: Time-agnostic reward, T: Time-based reward) vs **GPTSwarm*.** Evaluated on the test set by F1-score and time (log-transformed, in ms).

* For GPTSwarm, the final optimized graph configuration is used for evaluation.

GPTSwarm: Language Agents as Optimizable Graphs (Zhuge et al., 2024)

Designs of the future GenIA Systems should be;

Modular	Composed of heterogenous modules; agents, models and tools
Adaptive	To user input requirement to self-organize its architecture on-the-fly
Dynamic	During execution, to adapt to intermediate outcomes

Paper & Code

Adaptive Orchestration of Modular Generative Information Access Systems

Mohanna Hoveyda, Harrie Oosterhuis, Arjen P. de Vries, Maarten de Rijke, Faegheh Hasibi

IN·DEI·NOMINE·FELICITER

Thank you for listening!

Questions ?