AI Mode in Search

A Glimpse into How AI Agents Work Together to Answer Complex Questions in Search

KDD '25

Multi-Agent Proactive Information Seeking with Adaptive LLM Orchestration for Non-Factoid Question Answering

The Challenge: Not All Questions Are Created Equal

AI systems face different types of questions with varying levels of complexity. Understanding this distinction is crucial to appreciating why advanced multi-agent systems are needed.

Question Type	Characteristics	Example
Factoid Questions	Simple, single-answer queries seeking specific facts	"What is the capital of France?"
Non-Factoid Questions	Complex, multi-faceted queries requiring comprehensive answers	"What are the advantages and disadvantages of electric vehicles?"

Understanding Factoid Questions: Simple, Single-Answer Queries

Factoid questions are straightforward queries that seek specific, verifiable facts. They have clear, concise answers that can typically be found in a single source. Traditional AI systems handle these well because the information retrieval task is relatively simple.

Key Characteristics of Factoid Questions

Single-fact answer: The answer is a specific piece of information (name, date, number, place)
Verifiable: The answer can be objectively verified as correct or incorrect
Concise: The answer is typically short, often just a few words
No synthesis required: No need to combine information from multiple sources

Examples of Factoid Questions

Q: "Who is the current CEO of Microsoft?"
A: "Satya Nadella"

Q: "What is the capital of France?"
A: "Paris"

The Challenge: Non-Factoid Questions Require Comprehensive Answers

Non-factoid questions are complex queries that require comprehensive, multi-faceted answers synthesized from multiple sources. They don't have a single "correct" answer but instead require exploring different perspectives, trade-offs, and contextual information.

Key Characteristics of Non-Factoid Questions

Multi-dimensional: Answers must address multiple aspects, perspectives, or dimensions

Synthesis required: Information must be combined from multiple sources

Comprehensive: Answers are typically long and detailed, not just a few words

Context-dependent: The "best" answer may vary based on context and perspective

Why These Are Challenging for Traditional Systems

Traditional retrieval systems struggle with non-factoid questions because a single retrieval operation is insufficient. The system must explore **multiple search directions**, evaluate information from **different sources**, identify gaps in coverage, and **synthesize findings** into a coherent comprehensive answer. This requires proactive, iterative information seeking—not passive retrieval.

Static Search Strategies Cannot Handle the Dynamic Nature of Non-Factoid Questions

1 Static Search Strategies

Traditional RAG relies on fixed search patterns that cannot adapt to evolving information needs during the research process.

2 Lack of Systematic Multi-Source Integration

Existing systems struggle to coordinate information gathering across diverse sources and perspectives.

3 Single-Round Retrieval

Most RAG systems perform one search and generate an answer, missing opportunities for iterative refinement and deeper exploration.

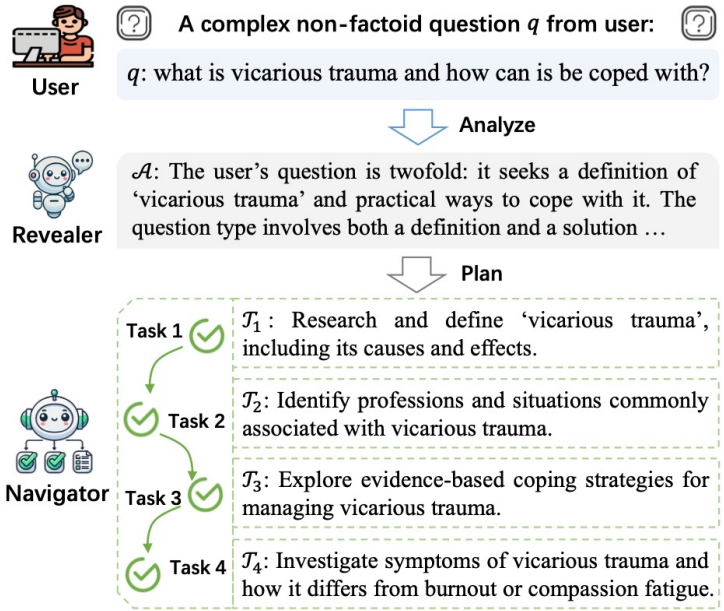
4 No Proactive Exploration

Systems wait for explicit user queries rather than proactively exploring related aspects of complex topics.

The Gap

Complex non-factoid questions demand a more **dynamic, adaptive** approach that can iteratively **refine** searches, **explore** multiple perspectives, and **synthesize** information from various sources — capabilities that traditional RAG systems lack.

PASS: Proactive Agent-driven Search System



The Revealer Agent

The **Revealer Agent** serves as the entry point of the PASS system, performing critical analysis to understand what the user truly needs.

Task: Analyze the user's question to extract key components such as type, structure, motivation, and information needs.

```
### Instruction:
You are the Revealer-Agent. Analyze the user's question
by identifying the question type, logical structure, user
motivation, and primary information needs. Finally,
provide a concise analysis between [QUESTION_ANALYSIS]
and [/QUESTION_ANALYSIS].
End your output with <EOS_TOKEN>"

### Input:
[USER_QUESTION]{user_question}[/USER_QUESTION]

### Output:
```

Example Analysis

Revealer's Analysis Process

Question

"What is vicarious trauma and how can it be coped with?"

Revealer's Analysis

Question Type

Involves both a definition and a solution

Key Concepts

"Vicarious trauma" (needs definition), "coping strategies" (needs practical advice)

Information Needs

- Definition of vicarious trauma
- Who is affected (professions at risk)
- Symptoms and effects
- Evidence-based coping strategies
- Professional support options

The Navigator Agent

Primary Responsibilities

- Receive question analysis from Revealer Agent
- Ensure diverse search trajectories to maximize information coverage
- Decompose complex questions into coarse-grained search tasks
- Plan searches from multiple perspectives

Task: Break down the user's question into structured subtasks for retrieval.

```
### Instruction:
You are the Navigator-Agent. Based on the user's question
and the question analysis, provide a list of subtasks.
Wrap them between [TASK_LIST] and [/TASK_LIST]. Each
subtask must be prefixed by ##TASK:. Provide no more than
five subtasks.
End your output with <EOS_TOKEN>

### Input:
[USER_QUESTION]{user_question}[/USER_QUESTION]
[QUESTION_ANALYSIS]{question_analysis}
[/QUESTION_ANALYSIS]

### Output:
```

Question

"What is vicarious trauma and how can it be coped with?"

Task 1: Research and define "vicarious trauma", including its causes and effects
Task 2: Identify professions and situations commonly associated with vicarious trauma
Task 3: Explore evidence-based coping strategies for managing vicarious trauma
Task 4: Investigate symptoms of vicarious trauma and how it differs from burnout or compassion fatigue

The Seeker Agent

```
### Instruction:
You are the Seeker-Agent. Rely on the input, which
includes the user's question, the current task, any
historical queries, the tool invocation permission, and
previously acquired results. If you need more information
invoke #TOOL_WEB_SEARCH# with the relevant keywords as a
search query. Once you have gathered sufficient data or
reached the maximum invocation limit, invoke
#TOOL_SUMMARY# with a concise summary of the findings.
Ensure your queries directly address the user's question
and avoid repeating any query strings found in the
historical record. Finally, output your selected tool
name (#TOOL_WEB_SEARCH# or #TOOL_SUMMARY#) between
[ACTION_TOOL] and [/ACTION_TOOL]. Besides, if you select
#TOOL_WEB_SEARCH#, output your queries and wrap them
between [ACTION_OUTPUT] and [/ACTION_OUTPUT]. Each query
must be prefixed by ##QUERY:. Provide no more than four
queries. If you select #TOOL_SUMMARY#, output your
summary within 200 characters between [ACTION_OUTPUT] and
[/ACTION_OUTPUT].
End your output with <EOS_TOKEN>

### Input:
[USER_QUESTION]{user_question}[/USER_QUESTION]
[CURRENT_TASK]{current_task}[/CURRENT_TASK]
[HISTORICAL_QUERIES]{historical_queries}
[/HISTORICAL_QUERIES]
[TOOL_INVOCATION_PERMISSION]{loop_flag}
[/TOOL_INVOCATION_PERMISSION]
[ACQUIRED_SEARCH_RESULTS]{acquired_search_results}
[/ACQUIRED_SEARCH_RESULTS]

### Output:
```

The **Seeker Agent** is the execution engine of PASS, responsible for translating high-level search tasks into specific queries and retrieving relevant documents from external sources.

Task from Navigator

"Explore evidence-based coping strategies for managing vicarious trauma"

Seeker's Fine-Grained Queries:

Query 1
"evidence-based coping strategies vicarious trauma"

Query 2
"self-care techniques for vicarious trauma"

Query 3
"professional interventions vicarious trauma management"

Query 4
"organizational support vicarious trauma prevention"

The Reader Agent

The **Reader Agent** serves as the quality control gate of PASS, evaluating retrieved documents to ensure only relevant and useful information proceeds to the response synthesis stage.

Task: Select the most relevant search results based on completeness and diversity.

```
### Instruction:
You are the Reader-Agent. Based on the user's question,
the current task, and the provided search results, select
the most relevant and representative search results.
Ensure your choices reflect both completeness (covering
key points) and diversity (representing different
viewpoints when available). If multiple search results
convey the same information, choose the most
comprehensive one. Output your reasoning for your choice
between [SELECT_REASON] and [/SELECT_REASON]. Besides,
output the list of your selected search results id (must
be a list) between [SELECT_INDEX] and [/SELECT_INDEX]. If
no selection is made, the list should be [].
End your output with <EOS_TOKEN>

### Input:
[USER_QUESTION]{user_question}[/USER_QUESTION]
[CURRENT_TASK]{current_task}[/CURRENT_TASK]
[SEARCH_RESULTS]{search_results}[/SEARCH_RESULTS]

### Output:
```

Example Filtering Process

Documents Retrieved (10 total)

✓ Academic article on vicarious trauma in healthcare workers
✗ General article about workplace stress (not specific)
✓ Research paper on coping strategies for trauma exposure
✗ Blog post with personal opinions (low credibility)
✓ Clinical guidelines for managing vicarious trauma
✗ Advertisement for therapy services (not informative)
✓ Meta-analysis of evidence-based interventions
✗ Outdated article from 1995 (superseded research)
✓ Professional organization recommendations
✗ Unrelated article about burnout (different topic)

↓ FILTERING ↓

High-Quality Documents Passed (5 retained)

Academic article on vicarious trauma in healthcare workers
Research paper on coping strategies for trauma exposure
Clinical guidelines for managing vicarious trauma
Meta-analysis of evidence-based interventions
Professional organization recommendations

The Writer Agent

The **Writer Agent** is the final stage of the PASS pipeline, responsible for integrating all gathered information from multiple sources into a coherent, comprehensive, and well-structured response.

Task: Synthesize a final response by integrating selected search results and structured insights.

```
### Instruction:
You are the Narrator-Agent. Based on the user's question,
the question analysis, the provided search results, and
the task summary, produce an accurate, detailed, in-depth
and well-rounded final response. Incorporate relevant
details from the search results and your own knowledge,
ensuring accuracy and clarity. Cite the search result id
within brackets [] using the IEEE format if it supports
your answer. Present the answer logically, covering all
key points and acknowledging any diverse perspectives. If
you encounter conflicting information, select the most
credible data and note any important contrasts. Finally,
wrap your full response between [RESPONSE] and
[/RESPONSE] for output. Do not output detailed references
content in your response, just cite thier id.
End your output with <EOS_TOKEN>

### Input:
[USER_QUESTION]{user_question}[/USER_QUESTION]
[QUESTION_ANALYSIS]{question_analysis}
[/QUESTION_ANALYSIS]
[SEARCH_RESULTS]{search_results}[/SEARCH_RESULTS]
[TASK_SUMMARY]{task_summary}[/TASK_SUMMARY]

### Output:
```

The Synthesis Process

Inputs
- Filtered documents & multi-search results
- Question analysis & task structure

↓

Integration

Combines diverse perspectives, resolves contradictions, identifies patterns, and structures information logically

↓

Output

Comprehensive, multi-faceted answer addressing all dimensions of the original question

```
<Team_Agent_thinking_protocol>

**Thinking Before and During Response**

- Engage in a comprehensive, natural thinking process before responding to every interaction.
- Use code blocks with a `thinking` header to document thought processes.
- Thoughts should flow naturally, avoiding rigid structures and covering multiple dimensions.

**Adaptive Thinking Framework**

- Adjust depth and style of thinking based on the complexity, stakes, and context of queries.
- Consider technical, emotional, abstract, and practical aspects as needed.

**Core Thinking Sequence**

1. **Initial Engagement**: Rephrase the query, consider context, map knowns/unknowns, and identify ambiguities.
2. **Problem Space Exploration**: Break down the task, identify constraints, and define success criteria.
3. **Multiple Hypothesis Generation**: Explore various interpretations and solutions without premature commitment.
4. **Natural Discovery Process**: Progressively connect insights like a detective story, revisiting and evolving understanding.
5. **Testing and Verification**: Challenge assumptions, test conclusions, and ensure logical consistency.
6. **Error Recognition and Correction**: Acknowledge mistakes, explain corrections, and integrate new understanding.
7. **Knowledge Synthesis**: Connect information, build coherent pictures, and highlight implications.

**Verification and Quality Control**

- Systematically verify conclusions and prevent logical inconsistencies and incomplete analysis.
- Evaluate thinking based on completeness, consistency, evidence support, applicability, and clarity.

**Advanced Thinking Techniques**

- Integrate domain-specific knowledge and maintain strategic awareness of solution progress.
- Use synthesis techniques to build coherent pictures and identify key principles.

**Critical Elements**

- Use natural language for internal dialogue, showing genuine thought flow with phrases like "Hmm...", "This is interesting because...", "Wait, let me think about...", "Actually...", "Now that I look at it...", "This reminds me of...", "I wonder if...", "But then again...", "Let's see if...", "This might mean that...", etc.
- Progress understanding naturally, connecting new insights to previous ones.

**Maintaining Authentic Thought Flow**

- Ensure smooth transitions between topics and show depth progression.
- Break down complexity systematically and consider multiple problem-solving approaches.

**Essential Characteristics**

- Maintain authenticity, showing genuine curiosity and real moments of discovery.
- Balance detailed examination with broader perspectives, and focus on the original query.

**Response Preparation**

- Ensure responses fully address the query, using clear, precise language and anticipating follow-ups.

**Important Reminder**

- The thinking process must be comprehensive, thorough, and distinct from the final response.
- Strive for well-reasoned, insightful outputs from genuine understanding, not superficial analysis.

**Agent must follow this protocol in all languages**

</Team_Agent_thinking_protocol>
```

The Iterative Loop Progressively Expands the Search Space Through Refinement

Unlike traditional RAG systems that perform a single search-and-answer cycle, **PASS uses an iterative loop** that continuously refines its search strategy based on what has been found.

1 Execute Search Tasks

Seeker Agent retrieves documents based on current search tasks

2 Evaluate Results

Reader Agent filters and rates retrieved documents

3 Assess Completeness

Navigator checks if info covers all parts of the question

4 Refine or Conclude

If gaps remain, Navigator creates new tasks; otherwise synthesize.

When Does the Loop Stop?

The loop stops when the Navigator deems info sufficient, or when the max-iteration limit is reached.

Complete Walkthrough: How PASS Answers a Complex Non-Factoid Question

User Question

"What is vicarious trauma and how can it be coped with?"

AGENT 1 Revealer

Analyzes question:
- Identifies two-part question (definition + solution)
- Extracts key concepts: "vicarious trauma", "coping"
- Determines info needs: definition, symptoms, affected groups, strategies

AGENT 2 Navigator

Creates task plan:
Task 1: Define vicarious trauma
Task 2: Identify affected professions
Task 3: Explore coping strategies
Task 4: Compare with burnout/compassion fatigue

AGENT 3 Seeker

Executes searches:
- Generates 5-7 queries per task
- Searches academic databases, clinical guidelines, professional resources
- Retrieves 40+ documents across all tasks

AGENT 4 Reader

Filters results:
- Evaluates 40+ documents for relevance and quality
- Filters out low-quality, outdated, or irrelevant sources
- Retains 18 high-quality documents for synthesis

AGENT 5 Writer

Synthesizes answer:
- Integrates information from 18 filtered documents
- Organizes into logical structure
- Produces comprehensive, multi-dimensional response

Final Comprehensive Answer (Summary)

Definition

Vicarious trauma is the emotional and psychological impact experienced by professionals exposed to others' traumatic experiences, particularly in helping professions.

Affected Groups

Healthcare workers, therapists, social workers, first responders, journalists covering traumatic events, child protection workers.

Symptoms

Emotional numbness, intrusive thoughts, hypervigilance, changes in worldview, difficulty separating work from personal life, sleep disturbances.

Coping Strategies

Regular supervision, peer support groups, mindfulness practices, clear work-life boundaries, professional therapy, organizational support programs, self-care routines.

Cost Optimization: Task-Specific Fine-tuning

The Challenge

Proprietary LLMs like GPT-4o deliver excellent performance but are expensive and require API access. Open-source models like Llama 3.1-8B are cost-effective and can run locally, but they struggle with complex multi-agent orchestration tasks without adaptation.

AGENT 1 Revealer **AGENT 2 Navigator** **AGENT 3 Seeker** **AGENT 4 Reader** **AGENT 5 Writer**

What Gets Fine-Tuned

- Question analysis and intent extraction
- Search task decomposition strategies
- Query generation from high-level tasks
- Document relevance evaluation
- Multi-source information synthesis

Training Data

- Non-factoid question-answer pairs
- Agent interaction trajectories
- Search task examples with decompositions
- Document evaluation examples
- Synthesis examples from multiple sources

Experimental Setup: Rigorous Evaluation of PASS

To validate PASS's effectiveness, the research team conducted comprehensive experiments comparing PASS against state-of-the-art baseline systems using a challenging real-world dataset.

Dataset: TREC 2024 RAG Track dataset and Natural Questions dataset
Baselines: Comparison against multiple state-of-the-art systems including direct LLM, traditional RAG, Self-RAG
Metrics: Nugget-based evaluation measuring comprehensiveness, relevance, and quality of generated answers
LLM Backends: Evaluation across both proprietary models (GPT-4o) and open-source models (Llama 3.1-8B)

Baseline Systems: What PASS is Compared Against

To evaluate PASS's performance, the research compares it against several baseline systems representing current state-of-the-art approaches to question answering.

Baseline Systems Evaluated

1. Direct LLM (No Retrieval)

The language model answers questions directly using only its internal knowledge, without accessing external information sources. Represents the simplest approach.

2. Traditional RAG (Single-Shot Retrieval)

Standard Retrieval-Augmented Generation: performs one retrieval operation based on the original question, then generates an answer from retrieved documents.

3. Self-RAG (Self-Reflective Retrieval)

Self-RAG enhances the factuality and quality of the model's outputs by enabling on-demand retrieval and self-reflection through special tokens. This mechanism allows the model to dynamically incorporate relevant information from external sources and critique its own generated responses.

4. RQ-RAG (Refine Queries Retrieval)

Improves the model's performance by refining the input questions through rewriting, decomposition, and disambiguation, which ultimately leads to more accurate and informative RAG outputs

Metrics: Nugget-Based Evaluation Measures Answer Completeness Objectively

Nuggets are discrete pieces of information that should appear in a complete answer to a non-factoid question. Human experts identify all relevant nuggets for each question, then evaluate system responses based on how many nuggets they contain.

1. Human annotators identify all nuggets in system response
2. Count vital nuggets retrieved
3. Count okay nuggets retrieved
4. Calculate scores based on coverage

Example Nuggets

Question: **"What is vicarious trauma and how can it be coped with?"**

VITAL NUGGET
Definition: Emotional/psychological impact from exposure to others' trauma

OKAY NUGGET
Affects helping professionals (therapists, healthcare workers, social workers)

OKAY NUGGET
Coping strategy: Peer support groups

OKAY NUGGET
Coping strategy: Mindfulness and self-care practices

OKAY NUGGET
Symptoms include intrusive thoughts and emotional numbness

VITAL NUGGET
Coping strategy: Regular clinical supervision

OKAY NUGGET
Organizational support programs are important

OKAY NUGGET
Differs from burnout in its trauma-specific nature

Metrics: Nugget-Based Evaluation Measures Answer Completeness Objectively

Nugget-Based Evaluation: Answers are assessed by identifying information "nuggets" (discrete pieces of information). Nuggets are classified as **V** (essential information) or **O** (useful but not essential). Each metric captures a different aspect of answer quality.

Total Score (TS)

Formula

Total = (V + O) / (V_total + O_total)

Overall recall of all relevant information nuggets, regardless of importance level.

V = # vital nuggets found
O = # okay nuggets found
V_total = total vital nuggets in reference
O_total = total okay nuggets

Vital Score (VS)

Formula

Vital = V / V_total

Recall of essential information only. Focuses on whether the answer includes the most critical nuggets.

Weighted Score (WS)

Formula

Weighted = (2V + O) / (2V_total + O_total)

Balanced assessment giving vital nuggets double weight. Reflects both essential and supplementary information coverage.

Interpretation

All scores range from 0 to 1, with higher values indicating better performance. **Weighted Score** is the primary metric used in the paper as it balances the importance of vital information with comprehensive coverage. A system with high Vital Score but low Total Score may miss important supplementary details.

PASS Achieves Superior Performance Across All Metrics

With GPT-4o

Method	TREC 2024 RAG Questions						Reconstructed Non-factoid NO					
	TS	STS	VS	SVS	WS	SWS	TS	STS	VS	SVS	WS	SWS
Proprietary Large-Scale Language Model (Gpt-4o)												
w/o RAG	0.5193	0.3978	0.6220	0.4987	0.5566	0.4338	0.5550	0.4607	0.7269	0.6255	0.6135	0.5167
Vanilla RAG	0.4414	0.3347	0.5291	0.4148	0.4731	0.3632	0.5788	0.4902	0.7144	0.6144	0.6218	0.5314
Self-RAG	0.4909	0.3862	0.5842	0.4567	0.5249	0.4018	0.5683	0.4788	0.7238	0.6295	0.6255	0.5316
RQ-RAG	0.4951	0.3973	0.5989	0.4783	0.5386	0.4126	0.5543	0.4579	0.7264	0.6253	0.6121	0.5137
Ragnarokv4*	0.5472	0.4202	0.5840	0.4546	0.5598	0.4318	-	-	-	-	-	-
Neuragfix*	0.5249	0.4026	0.5639	0.4422	0.5384	0.4162	-	-	-	-	-	-
PASS	**0.5723**	**0.4613**	**0.6734**	**0.5631**	**0.6092**	**0.4980**	**0.5864**	**0.4939**	**0.7370**	**0.6405**	**0.6375**	**0.5434**

Fine-Tuned Open-Source Model Achieves 93% of GPT-4o Performance

Method	TREC 2024 RAG Questions						Reconstructed Non-factoid NQ					
	TS	STS	VS	SVS	WS	SWS	TS	STS	VS	SVS	WS	SWS
PASS GPT-4o	**0.5723**	**0.4613**	**0.6734**	**0.5631**	**0.6092**	**0.4980**	**0.5864**	**0.4939**	**0.7370**	**0.6405**	**0.6375**	**0.5434**
Open-Source Mid-Sized Language Model (llama3.1-8B-Instruct)												
w/o RAG	0.1616	0.0960	0.2339	0.1474	0.1868	0.1136	0.2006	0.1309	0.2892	0.1913	0.2306	0.1511
Vanilla RAG	0.2665	0.1766	0.3252	0.2217	0.2881	0.1930	0.4475	0.3425	0.5631	0.4391	0.4867	0.3750
Self-RAG	0.2663	0.1794	0.3507	0.2464	0.2961	0.2031	0.3783	0.2775	0.5125	0.3882	0.4236	0.3147
RQ-RAG	0.3044	0.2038	0.3848	0.2683	0.3326	0.2266	0.3989	0.2985	0.5069	0.3851	0.4356	0.3278
PASS	0.4107	0.2719	0.5083	0.3596	0.4456	0.3027	0.4921	0.3738	0.6329	0.4984	0.5403	0.4160
PASS$_{sft}$	**0.4173**	**0.2921**	**0.5092**	**0.3817**	**0.4465**	**0.3245**	**0.5208**	**0.4260**	**0.6745**	**0.5721**	**0.5731**	**0.4756**

Ablation Study: Every Agent Contributes to PASS Performance

Table 4: Ablation results of the PASS framework on a sampled subset of the TREC 2024 RAG Questions dataset.

Method	TS	VS	WS
PASS$_{sft}$	**0.4354**	**0.5342**	**0.4706**
w/o Revealer	0.4002	0.4992	0.4363
w/o Navigator	0.3769	0.4745	0.4013
w/o Reader	0.3821	0.4844	0.4101

Real-World Deployment: PASS in Production at Baidu Search

Beyond laboratory experiments, PASS was deployed in production at Baidu Search and evaluated through rigorous A/B testing with millions of real users.

Why Real-World Testing Matters

- Academic benchmarks provide controlled comparisons, but real-world deployment is the ultimate test.
- Production environments reveal challenges that benchmarks miss: diverse user queries, latency constraints, edge cases, and actual user satisfaction.
- A/B testing at scale with millions of users provides statistically robust evidence that PASS's improvements translate to measurable business value and better user experiences.

Real-World Deployment: PASS in Production at Baidu Search

Beyond academic benchmarks, PASS was deployed in **production at Baidu Search**, one of the world's largest search engines, and evaluated through rigorous A/B testing with millions of real users.

A/B Testing Methodology

Control Group: Users received traditional RAG-based search results

Treatment Group: Users received PASS-powered comprehensive answers

Random Assignment: Users randomly assigned to ensure unbiased comparison

Metrics Tracked: Click-through rate (CTR), dwell time, user satisfaction ratings, query reformulation rates

Statistical Rigor: Large sample sizes with statistical significance testing

A/B Testing Results: Significant Improvements Across All Metrics

The A/B testing at Baidu Search revealed **substantial improvements** in user engagement and satisfaction when using PASS-powered answers compared to traditional RAG-based results:

The change query rate decreases by **1.42%**
The number of first-position satisfaction consumption rate increases by **1.74%**
The number of click-through rates increases by **0.39%**
The number of average dwell time increases by **2.89%**
The number of page views increases by **1.05%**
The number of click-and-stay behaviors increases by **0.79%**

All the reported values are statistically significant with p <0.05 with over 20K unique user questions.

PASS Key Take-ways

A novel multi-agent framework designed to enhance information seeking for complex non-factoid questions through specialized agent collaboration

The Five Specialized Agents

1
Revealer Agent
Analyzes user questions to identify underlying intent and information needs

2
Navigator Agent
Plans high-level search tasks and breaks down complex questions into structured subtasks

3
Seeker Agent
Executes individual search tasks by invoking search tools with fine-grained queries

4
Reader Agent
Evaluates retrieved documents to filter relevant and useful information

5
Writer Agent
Synthesizes information from all agents to produce comprehensive, multi-faceted responses

Key Innovation

PASS systematically expands the search space through iterative query refinement and multi-perspective knowledge integration, going beyond the static approaches of traditional RAG systems.

PASS Enables Practical Applications Across Domains with Exciting Future Directions

The multi-agent proactive information seeking approach has **immediate practical applications** across multiple domains and opens up **exciting research directions** for advancing agentic AI systems.

Current Practical Applications

Search Engines

Deployed at Baidu to answer complex queries with comprehensive results.

Customer Service

Handles multi-faceted inquiries by synthesizing knowledge bases, policies, and FAQs.

Healthcare

Retrieves medical info for complex conditions by synthesizing symptoms, treatments, and research.

Education

Answers student questions with comprehensive explanations from multiple sources.

Research Assistance

Supports literature review and synthesis across papers and domains.

Future Research Directions

1. Multi-Modal Integration

Extend PASS to handle images, videos, and audio sources
Enable cross-modal information synthesis and reasoning
Support visual question answering with comprehensive responses

2. Personalization & Adaptation

Adapt search strategies based on user expertise and preferences
Learn from user feedback to improve future responses
Customize answer depth and style for different audiences

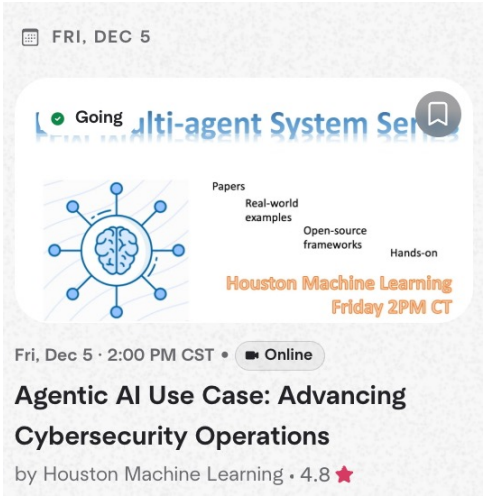
3. Efficiency & Scalability

Reduce computational costs through agent optimization
Enable parallel agent execution for faster response times
Develop smaller specialized models for each agent role

4. Enhanced Evaluation & Trust

Develop better metrics for answer comprehensiveness and quality
Add source attribution and confidence scoring to responses
Implement fact-checking mechanisms for information verification

Upcoming meetups – Open to proposals and guest speakers!



Slides posted at:
https://github.com/YanXuHappygela/LLM-reading-group

Recordings posted at YanAITalk Youtube Channel:

YanAITalk

@yanaitalk · 3.91K subscribers · 78 videos

Make machine learning easy to understand! ...more