New Architectures, Cognitive Strategies, and Validation Protocols in Deep Research Agent Development: A Comprehensive Technical and Methodological Analysis

Chapter 1: The Paradigm Shift in AI: Transitioning from Language Models to Autonomous Agentic Systems

In the history of artificial intelligence development, the emergence of "Deep Research Agents" is recognized as a fundamental turning point that has shattered the boundaries between static natural language processing and dynamic problem-solving. While Large Language Models (LLMs) traditionally functioned as text prediction engines guessing the next word based on statistical probabilities, modern agents represent an evolutionary leap toward systems possessing "Agency." This transition from the "Reactive" nature of traditional software—which merely waits for user input to execute a specific command—to the "Proactive" nature of agents that understand abstract goals and autonomously move toward realizing them, has created a fundamental shift in the philosophy of human-machine interaction.[1]

1.1. The Nature of Deep Research Agents

A deep research agent, unlike a simple chatbot, is designed to simulate the complex cognitive processes of a human researcher. These systems not only have access to knowledge bases but also possess the ability for "Long-horizon Reasoning." This capability allows them to manage research tasks that may take hours or even days, refine their search strategies based on intermediate findings, and ultimately transform a volume of scattered information into coherent, structured knowledge.[3] The key difference here is "Context Retention" and "State Management," where the agent must maintain a continuous train of thought across thousands of execution steps without succumbing to forgetfulness or deviating from the main objective.

1.2. The Three Pillars of Architectural Transformation

A review of recent technical literature and research reports reveals that the effectiveness of these agents rests on three main pillars that distinguish them from previous generation systems:

1. **From Tactics to Strategy:** In traditional models, the focus was on immediate execution

of a command. However, in deep research agents, the first action is "planning." Before any interaction with the environment or search tools, the system formulates a comprehensive strategy, often modeled as a Directed Acyclic Graph (DAG). This graph defines dependencies between various tasks and outlines the optimal path to reach the final goal.[1]

2. **From Sequential to Parallel Processing:** Modern agents act like project managers. Instead of performing all steps sequentially themselves, they distribute tasks among multiple "Specialist Agents." These sub-agents can simultaneously investigate different aspects of a problem; for example, while one agent examines financial data, another analyzes legal records.[1]
3. **From Scratchpad to Persisted State:** One of the biggest challenges of language models is the limitation of short-term memory (Context Window). Deep research agents use advanced state management architectures (such as LangGraph) to store information in a structured and persistent manner. This enables the collection, refinement, and synthesis of information over long time scales, allowing the system to provide analyses requiring deep, multi-layered understanding of the subject.[1]

This architectural transformation has not only increased problem-solving capacity but has also introduced new challenges in agent coordination, cost management, and quality assurance, which will be examined in detail in subsequent chapters.

Chapter 2: Cognitive Prompt Engineering and Task Decomposition Strategies

The success of an intelligent agent in executing complex tasks depends directly on the quality of input instructions and how its thinking process is structured. "Agentic Prompt Engineering" goes beyond choosing the right words and has become a form of "Cognitive Engineering." The main goal here is to break the inherent limitations of language models in multi-step reasoning through structured analysis techniques.

2.1. Artificial Cognitive Psychology and Reducing Cognitive Load

Similar to the human mind, language models have limitations in "Cognitive Load." When a model is asked to process multiple complex concepts, numerous variables, and logical constraints simultaneously, the probability of error, hallucination, and superficial reasoning increases dramatically. The technique of "Decomposed Prompting" (DecomP) has been proposed as a solution to this problem. By breaking a large problem into smaller, manageable sub-problems, the model can focus its entire processing power on a specific aspect.[5]

Research has shown that this focused approach leads to more reliable and accurate outputs. The decomposition process not only increases accuracy but also creates other systemic benefits:

- **Task Organization:** Creating a clear roadmap that allows progress tracking.
- **Modular Debugging:** If an error occurs, it can be traced and corrected in a specific sub-task without affecting the entire process.
- **Targeted Refinement:** The ability to review specific parts of the analysis without needing to repeat all calculations.[5]

2.2. Advanced Tactics in Reasoning Decomposition

To effectively implement this strategy, several methodologies have been developed, each suitable for a specific type of problem:

2.2.1. Chain of Thought (CoT)

Also known as Single-path Reasoning, this technique forces the model to articulate its intermediate reasoning steps before arriving at a final answer. CoT is highly effective for problems with a linear logical progression (such as math problems or deductive inferences).[6] However, in open-ended research problems lacking a clear path, CoT can be limiting as it follows only a single line of thought.

2.2.2. Tree of Thoughts (ToT)

To overcome the limitations of linear reasoning, the "Tree of Thoughts" technique was introduced. This is a Multi-path Reasoning approach where the model is encouraged to explore multiple branches of possibilities simultaneously. At each step, the model generates several possible options for the next move, evaluates them, and prunes branches that lead to dead ends or are of low quality.[6] This approach mirrors the thinking process of a researcher who may consider multiple hypotheses, reject some, and select the most promising ones for deeper investigation. In complex research environments, ToT enables the discovery of hidden connections and creative solutions often overlooked in linear methods.

2.2.3. Program Synthesis and Hierarchical Decomposition

In this advanced method, reasoning is expressed as a computer program where each step is defined as a function. This approach guarantees the highest level of precision and reproducibility.[5] Additionally, "Hierarchical Decomposition" introduces an abstract planning layer prior to execution. The model first designs the overall problem-solving structure (like a book's table of contents) and then delves into the details of each section. This separation of "design" from "execution" ensures the overall coherence of the final report and prevents the model from getting lost in details.[5]

2.3. Self-Criticism Mechanisms and Recursive Feedback

One of the most powerful tools in the agentic prompt engineering toolkit is "Self-Criticism." In this process, the model is asked to review its outputs as an unbiased critic. Self-criticism prompts guide the model to identify potential errors, logical gaps, or lack of evidence and correct them before finalizing the response.[8] This recursive loop significantly enhances output quality, as the model has the opportunity to revise its initial "draft," much like human writers do.

Combining these techniques—task decomposition, using tree structures for solution exploration, and applying multiple rounds of self-criticism—forms the cognitive foundation of deep research agents, enabling them to operate with a depth and precision previously out of reach for AI systems.

Chapter 3: Agentic Systems Architecture: Designing for Scalability and Persistence

Implementing the cognitive strategies discussed in the previous chapter requires a robust software foundation capable of managing the complexities of multi-agent interactions, data management, and operational stability. Agentic Systems Architecture serves as the skeleton of these intelligent entities.

3.1. Anatomy of a Deep Research Agent

A modern agentic system typically consists of several key modules interacting with one another:

1. **The Planner:** This is the strategic brain of the system. The planner is responsible for receiving the user query, performing initial analysis, and converting it into a comprehensive research plan. This plan is usually structured as a Directed Acyclic Graph (DAG) to manage dependencies and parallelization.[1]
2. **The Orchestrator:** This module handles executive management. The orchestrator is tasked with resource allocation, invoking sub-agents, managing data flow between modules, and monitoring the overall system status. Using central orchestration patterns (Central Master Agent) helps prevent chaos in multi-agent systems.[1]
3. **Worker Agents:** These are specialized units optimized for specific tasks. There might be a Searcher agent for interacting with search engines, a Reader agent for extracting information from long PDF files, and a Writer agent for generating final text. Using different language models for these roles (e.g., cheaper models for simple tasks and powerful models for reasoning) is a strategy for cost and performance optimization.[1]

3.2. State Management and Shared State

In deep research involving thousands of interaction steps, maintaining information continuity is vital. Modern architectures like LangGraph model the entire research workflow as a "Shared State." In this model, all agents write to and read from a central state repository regarding inputs, outputs, and intermediate data. This approach has several advantages:

- **Persistence:** System state can be checkpointed at any moment. This allows the system to resume exactly from where it left off after an error or need for human intervention, without repeating costly previous steps.[1]
- **Asynchronous Collaboration:** Different agents can work at different speeds and add their results to the shared state when ready, increasing overall system efficiency.

3.3. Parallelization Strategies and Dependency Graphs

One of the greatest advantages of intelligent agents over human researchers is their ability to perform parallel tasks. Using the DAG structure, the system can identify which parts of the research are independent and can be executed simultaneously. For example, in market research, competitor analysis, technological trend analysis, and regulatory review can run as independent, parallel branches. This parallelization not only drastically increases speed to result but also manages the information load on the main agent through "Compression" by sub-agents.[1] Each sub-agent is responsible for deep exploration in its domain and then summarizing the most important findings for the lead agent, enabling coverage of a vast breadth of information without overflowing the Context Window.

Chapter 4: Advanced Information Retrieval and Context Engineering

At the heart of every credible research process lies the ability to access accurate and relevant information. For intelligent agents, this process goes beyond simple keyword searching and enters the realm of "Context Engineering" and advanced Information Retrieval (IR) techniques.

4.1. Context Engineering: Beyond Prompting

While prompt engineering focuses on how to instruct the model, context engineering addresses optimizing the information environment in which the model makes decisions. This means carefully managing the tokens placed in the model's context window to ensure desired behavior.[10] The main challenge here is the model's memory limit and the "Lost in the Middle" phenomenon, where models tend to forget information located in the middle of a long text. Context engineering strategies include careful document selection, hierarchical summarization before entry into context, and structuring information into processable formats like JSON or Markdown tables so the model can better understand data relationships.[11]

4.2. Query Expansion Techniques

Human users often phrase their queries vaguely, incompletely, or using colloquial terms. For a research agent, relying on these raw inputs can lead to retrieval failure. To overcome this, "LLM-based Query Expansion" techniques are used.[12]

4.2.1. Hypothetical Document Embeddings (HyDE)

One innovative method in this area is "Hypothetical Document Embeddings" (HyDE). Instead of searching the user's question directly, the agent first attempts to generate a hypothetical answer (even if incorrect) to the question. This hypothetical answer is then vectorized and used to search for similar documents. The logic is that the hypothetical answer shares more lexical and semantic similarity with target documents than the user's question does. This method bridges the semantic gap between question and answer.

4.2.2. Corpus-Steered Expansion and MUGI

More advanced methods, such as "Corpus-Steered Query Expansion," use initial search results to extract technical terms and key sentences, combining them with the model's linguistic capabilities to construct new, highly precise queries.[12] Additionally, frameworks like MUGI (Multi-Text Generation Integration) ask the language model to generate multiple different versions of hypothetical references. Using adaptive weighting mechanisms, the impact of each reference on the search process is adjusted. This "multi-perspective" approach increases the probability of finding relevant documents even in highly specialized domains and reduces system reliance on a single attempt at query generation.[13]

4.3. Challenges in Retrieval-Augmented Generation (RAG)

Despite advancements, RAG systems still face serious challenges. "Information Conflict" is a major issue where different documents present contradictory information (detailed in the next chapter). Furthermore, "Lack of Domain Context" can lead to inappropriate query expansion; for instance, expanding the word "Virus" in medical research should differ from computer science research.[14] Advanced agents attempt to mitigate these ambiguities by analyzing the domain initially and utilizing specialized knowledge bases.

Chapter 5: Knowledge Synthesis, Multi-Document Reasoning, and Conflict Resolution

One of the most complex and critical stages in the operation of deep research agents is the Synthesis phase. Gathering information is only half the battle; the main challenge lies in combining this scattered, heterogeneous, and sometimes contradictory information to create a coherent, truth-oriented narrative. This chapter examines the cognitive mechanisms and architectures enabling agents to move beyond "summarization" to achieve "knowledge synthesis."

5.1. Inherent Challenges in Multi-Document Reasoning

Unlike single-document tasks where the model only needs to extract and paraphrase information from one text, Multi-Document Environments introduce a new level of complexity. In these environments, "Grounding" facts becomes more difficult because there is no single source of truth. Research shows that as the number of input documents increases, model "Hallucination" rates rise, and their ability to identify contradictions decreases.15
A classic example is semantic contradictions. Suppose one document says "Milk tea costs $10" and another says "Spend $20 to buy a cup of milk tea." While the numbers differ, one might be unit price and the other a minimum spend or bundle. Simple RAG models often fail to grasp these nuances, mistakenly flagging them as pure contradictions or, worse, merging them without explanation.16

5.2. Advanced Conflict Resolution Protocols

To empower agents to manage these complexities, researchers have developed specific methodologies:

5.2.1. Multi-Agent Debate Approach (MADAM-RAG)

One innovative solution is the "Debate" architecture. In the MADAM-RAG system, instead of a single agent trying to select the best answer, multiple agents with different perspectives or access to different document subsets engage in a debate. They present arguments, critique each other's evidence, and attempt to refute incorrect information. Finally, an "Aggregator Agent" reviews all arguments, filters out noise and misinformation, and synthesizes the final answer. Empirical results show this method significantly enhances the system's ability to handle ambiguity, misinformation, and noise simultaneously, improving performance over traditional RAG systems by up to 15%.[17]

5.2.2. Synthesis Matrix

To organize information at scale, agents use a technique similar to human researchers called a "Synthesis Matrix." In this method, the agent creates a grid structure (mental or actual) with information sources on one axis and key themes or questions on the other. The agent must fill the intersections with extracted evidence. This structure allows the agent to quickly identify patterns of "Agreement," "Disagreement," and "Information Gaps" across sources.[19] Using this matrix transforms report writing from blind copying into structured analysis, enabling the agent to generate analytical statements like "While Source A emphasizes X, Source B challenges it by presenting evidence Y."

5.3. Explicit Informing and Contradiction Classification

Research indicates that "Explicit Informing" the model about potential contradiction types improves performance. If the model knows it might encounter "Temporal Conflict" (old vs. new info) or "Definitional Conflict" (different terms), it adapts its reasoning strategies.[21] For example, when facing temporal conflict, the model automatically weighs newer sources more heavily. This requires a pre-processing layer where document metadata (e.g., publication date, publisher credibility) is extracted and provided as auxiliary signals to the model.

Chapter 6: Quality Assurance, Verification, and Bias Mitigation

In professional and academic applications, "Trust" is as valuable as "Intelligence." A research agent, no matter how smart, is unusable if prone to generating false or biased information. This chapter explores defense layers and control mechanisms designed to ensure the health and accuracy of agent outputs.

6.1. Hallucination Autopsy and Counter-Strategies

"Hallucination" in language models refers to the phenomenon where the model generates a response that seems grammatically and semantically correct but is factually wrong.22 This stems from the probabilistic nature of models trained to complete patterns, not seek truth.
To combat this in research agents, multi-layered solutions are used:

1. **Mandatory Citations and Source Verification:** Agents are required to provide precise citations for every claim. However, citations themselves can be hallucinations (e.g., the *Mata v. Avianca* legal case where AI fabricated court cases [23]). Therefore, advanced systems use a "Citation Verifier" layer that checks the existence of links and the alignment of content with the claim.
2. **Structured Formatting (XML Tagging):** Using XML tags in prompts (like <evidence>, <reasoning>, <conclusion>) helps the model separate boundaries between actual data and its own reasoning. This simple separation reduces hallucination rates by preventing the leakage of the model's general knowledge into the evidence section.[9]
3. **Cost-Effective Verification:** Since web searching to verify every sentence is costly and slow, modern approaches suggest using the internal knowledge of large models for initial verification and engaging in web searches only for suspicious or critical cases.[24]

6.2. Algorithmic Bias: Detection and Mitigation

Intelligent agents can reflect or even exacerbate biases present in their training data. These biases can include gender, racial, or cultural stereotypes.
Bias Mitigation strategies include:

- **Perspective Prompting:** Instructing the agent to examine a topic from multiple viewpoints (e.g., "Analyze this from the perspective of an economist, a sociologist, and an environmentalist"). This forces the model away from stereotypical responses.[25]
- **Warning Tokens:** Adding specific tokens to the prompt warning the model to act with caution regarding sensitive topics and activate self-censorship or moderation mechanisms.[26]
- **Fine-tuning with Human Feedback (RLHF):** Training the model with data explicitly

designed to reduce bias and increase fairness.

- **Risk of Overcorrection:** It must be noted that excessive efforts to reduce bias can lead to historical distortions (e.g., generating images of WWII soldiers with unreasonable racial diversity). Balancing "Fairness" and "Historical Accuracy" is a delicate challenge in system design.[27]

6.3. Automated Fact-Checking Frameworks

Advanced systems use frameworks like ReAct to create fact-checking feedback loops. In this cycle, the agent generates a claim, then automatically creates a "verification query," performs a search, and compares the result with the initial claim. Only if they match is the claim included in the final report. Although slower, this process raises accuracy standards to a level nearing human research.[28]

Chapter 7: Human-in-the-Loop (HITL) and Interaction Design

Despite impressive progress in agent autonomy, removing humans entirely from the decision loop is impossible and often undesirable. The "Human-in-the-Loop" (HITL) paradigm has emerged as an essential design pattern for deep research systems, balancing automation with human oversight.

7.1. Interaction Patterns and Control Points

Designing HITL systems requires architectures that allow for "Pause and Resume." Unlike simple Python scripts that run start-to-finish, a research agent must be able to stop at critical points and await user approval or correction.
Main patterns include:

- **Plan Approval:** After the "Planner" agent creates the research strategy (DAG), the system pauses and presents the roadmap to the user. The user can remove unnecessary branches, add new topics, or change priorities. This prevents wasted resources on wrong paths.[1]
- **Source Review:** In sensitive research, the agent may present a list of found sources or key papers to the user before deep reading to ensure validity and relevance.
- **Dead-end Intervention:** If the agent fails to find information for a sub-task or faces unsolvable contradictions, it can "Call for Help," asking the user to provide more context or steer the research path.[29]

7.2. Transparency and Trust

HITL is not just for error control; it is a key factor in building "Trust." When users see how the agent reasons and have intervention power at key points, they are more likely to accept final results. This approach relieves pressure on system designers to build "perfect" algorithms, as Edge Cases and unforeseen complexities can be managed by human judgment.[30]

7.3. Technical Infrastructure for HITL

Implementing these patterns requires complex state management infrastructure. Tools like Temporal or LangGraph enable "Checkpointing." This means saving the full agent memory state, variables, and execution history in a database. Thus, the system can wait hours or days for user input and then resume exactly from that point without losing context or needing reprocessing.[1] This feature is vital for research agents managing long-term projects.

Chapter 8: Benchmarks, Performance Evaluation, and Comparative Analysis

In data science, the saying goes: "You cannot improve what you cannot measure." Evaluating deep research agent performance is a unique challenge because traditional text metrics like ROUGE or BLEU, which focus on surface word similarity, are completely ineffective for measuring "Reasoning Depth," "Factual Accuracy," and "Structural Quality" of long reports.

8.1. Emergence of Agent-Specific Benchmarks

To address this need, the scientific community has developed novel benchmarks specifically designed to evaluate research processes:

8.1.1. ReportBench

This systematic benchmark is designed to evaluate the quality of reports generated by deep research agents. Its main focus is on two axes: reference quality and factual accuracy of all statements. ReportBench uses scientific survey papers from arXiv as "Ground Truth," comparing the agent's report against the structure and content of these expert papers.[32]

8.1.2. RACE and FACT Frameworks

These two frameworks work complementarily:

- **FACT (Framework for Factual Abundance and Citation Trustworthiness):** Focuses on the quantity and quality of extracted facts. It measures metrics like "Effective Citations" and "Citation Accuracy." Recent results show models like *Gemini-1.5-Pro Deep Research* achieving a score of 111.21 in effective citations, showing high power in information gathering, while *Perplexity Deep Research* offers the most accurate citations with 90.24% accuracy.[33]
- **RACE:** Focuses on qualitative report evaluation, using metrics like structural coherence, adherence to guidelines, and analytical depth.[33]

8.2. Performance Analysis and Key Findings

Reviews of leading models (such as OpenAI Deep Research, Gemini, Claude, etc.) have revealed interesting facts:

Table 1: Performance Comparison of Leading Models in Research Metrics

33

Model / Agent	Key Strengths	Weaknesses	Ideal Use Case
OpenAI Deep Research	Highest score in Instruction Following	High cost due to high token count	Precise structured reports based on client format
Gemini-1.5-Pro	Superior in Effective Citation volume and broad retrieval	Sometimes suffers from information scatter	Broad exploratory research requiring massive data collection
Perplexity Deep Research	Exceptional Citation Accuracy (90.24%)	Less analytical depth compared to competitors	Verification and research where source precision is critical
Kimi-K2	High raw quality of generated text	Less stability across different domains	Generating fluid and readable textual content

8.3. Cross-Domain Stability

An often overlooked but important metric is agent performance stability across different domains (medical, legal, financial, technical). Low standard deviation in performance scores across domains indicates a mature and generalized agent architecture. Data shows that while newer models may have high average scores, they may suffer severe performance drops in specific domains, whereas models like GPT-4 exhibit greater stability.[35]

Conclusion and Future Outlook

The comprehensive analysis of architectures, strategies, and challenges of deep research agents indicates we are at the beginning of a new era of knowledge production. These systems are no longer passive auxiliary tools but active cognitive partners capable of shouldering the heavy burden of primary research, information synthesis, and knowledge organization.

Key Findings and Strategic Recommendations

1. **Architecture Over Model:** The success of a research agent depends more on system architecture design (task decomposition, state management, interaction protocols) than on raw language model power. An average model with excellent architecture can outperform an excellent model with poor architecture.[9]
2. **Simplicity in Design:** Despite the theoretical appeal of complex adaptive systems, in practice, simple repeatable cycles of "Plan -> Execute -> Verify" yield the most stable results. Over-complexity often leads to unpredictable behaviors and debugging difficulties.[9]
3. **Focus on Synthesis, Not Just Retrieval:** The core value add of these agents is their ability to resolve contradictions and create new connections between discrete data. Investing in synthesis modules and multi-document reasoning (like MADAM-RAG) will yield the highest returns in final report quality.
4. **Transparency as a Necessity:** With increasing agent power and autonomy, the need for transparency mechanisms (like showing Chain of Thought, precise citations, and human intervention capability) is not just an ethical feature but an operational necessity to gain professional user trust.

Ultimately, the future of this field is moving toward "Collaborative Multi-Agent Ecosystems"; where specialized agents in different fields (e.g., a biology expert, a data analyst, and an ethics expert) collaborate to solve major scientific and societal challenges beyond the processing power of any single human or machine. This human-machine synergy will open new horizons in the speed and depth of scientific discovery.

End of Comprehensive Report

Works cited

1. Inside the Architecture of a Deep Research Agent - Egnyte Blog, accessed January 8, 2026, https://www.egnyte.com/blog/post/inside-the-architecture-of-a-deep-research-agent/
2. A practical guide to building agents - OpenAI, accessed January 8, 2026, https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-buil

ding-agents.pdf

3. Building Long-Running Deep Research Agents: Architecture, Attention Mechanisms, and Real-World Applications | by Madhur Prashant | Medium, accessed January 8, 2026, https://medium.com/@madhur.prashant7/building-long-running-deep-research-agents-architecture-attention-mechanisms-and-real-world-11f559614a9c
4. How we built our multi-agent research system - Anthropic, accessed January 8, 2026, https://www.anthropic.com/engineering/multi-agent-research-system
5. Break Down Your Prompts for Better AI Results, accessed January 8, 2026, https://relevanceai.com/prompt-engineering/break-down-your-prompts-for-better-ai-results
6. LLM Agents - Prompt Engineering Guide, accessed January 8, 2026, https://www.promptingguide.ai/research/llm-agents
7. How task decomposition and smaller LLMs can make AI more affordable - Amazon Science, accessed January 8, 2026, https://www.amazon.science/blog/how-task-decomposition-and-smaller-llms-can-make-ai-more-affordable
8. Building Effective Prompt Engineering Strategies for AI Agents - DEV Community, accessed January 8, 2026, https://dev.to/kuldeep_paul/building-effective-prompt-engineering-strategies-for-ai-agents-2fo3
9. How a deep research agent was built (and why simple workflows beat “smart” ones) - Reddit, accessed January 8, 2026, https://www.reddit.com/r/AI_Agents/comments/1ndkqxc/how_a_deep_research_agent_was_built_and_why/
10. Effective context engineering for AI agents - Anthropic, accessed January 8, 2026, https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents
11. Context Engineering: Moving Beyond Prompting in AI - DigitalOcean, accessed January 8, 2026, https://www.digitalocean.com/community/tutorials/context-engineering-moving-beyond-prompting-ai
12. LLM-Based Query Expansion Methods - Emergent Mind, accessed January 8, 2026, https://www.emergentmind.com/topics/llm-based-query-expansion
13. Exploring the Best Practices of Query Expansion with Large Language Models - Liner, accessed January 8, 2026, https://liner.com/review/exploring-best-practices-query-expansion-with-large-language-models
14. Query Expansion in Enhancing Retrieval-Augmented Generation (RAG) - Medium, accessed January 8, 2026, https://medium.com/@sahin.samia/query-expansion-in-enhancing-retrieval-augmented-generation-rag-d41153317383
15. Beyond the Single Document: Advancing Multi-Document Reasoning with LLMs, accessed January 8, 2026,

https://megagonlabs.medium.com/beyond-the-single-document-advancing-multi-document-reasoning-with-llms-2d24ae5e85bd

16. Seeking Advice: LLM Struggles to Find Contradictions Across Multiple Documents, accessed January 8, 2026, https://community.deeplearning.ai/t/seeking-advice-llm-struggles-to-find-contradictions-across-multiple-documents/877982
17. Retrieval-Augmented Generation with Conflicting Evidence - OpenReview, accessed January 8, 2026, https://openreview.net/forum?id=z1MHB2m3V9
18. Retrieval-Augmented Generation with Conflicting Evidence - arXiv, accessed January 8, 2026, https://arxiv.org/html/2504.13079v2
19. How to Synthesize Information from Multiple Sources: Techniques for Combining Different Perspectives, accessed January 8, 2026, https://www.sourcely.net/post/how-to-synthesize-information-from-multiple-sources
20. Synthesizing Sources | Examples & Synthesis Matrix - Scribbr, accessed January 8, 2026, https://www.scribbr.com/working-with-sources/synthesizing-sources/
21. (D)RAGged Into a Conflict: Detecting and Addressing Conflicting Sources in Retrieval-Augmented LLMs - Google Research, accessed January 8, 2026, https://research.google/pubs/dragged-into-a-conflict-detecting-and-addressing-conflicting-sources-in-retrieval-augmented-llms/
22. Hallucination (artificial intelligence) - Wikipedia, accessed January 8, 2026, https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence)
23. When AI Gets It Wrong: Addressing AI Hallucinations and Bias, accessed January 8, 2026, https://mitsloanedtech.mit.edu/ai/basics/addressing-ai-hallucinations-and-bias/
24. The cost of truth: An efficient fact-checking framework | NAACL - MBZUAI, accessed January 8, 2026, https://mbzuai.ac.ae/news/the-cost-of-truth-an-efficient-fact-checking-framework-naacl/
25. How to avoid replicating bias and human error in LLMs - Hello Future, accessed January 8, 2026, https://hellofuture.orange.com/en/how-to-avoid-replicating-bias-and-human-error-in-llms/
26. Understanding and Mitigating the Bias Inheritance in LLM-based Data Augmentation on Downstream Tasks - arXiv, accessed January 8, 2026, https://arxiv.org/html/2502.04419v2
27. Five strategies to mitigate bias when implementing generative AI - TELUS Digital, accessed January 8, 2026, https://www.telusdigital.com/insights/data-and-ai/article/mitigating-genai-bias
28. The perils and promises of fact-checking with large language models - PMC - NIH, accessed January 8, 2026, https://pmc.ncbi.nlm.nih.gov/articles/PMC10879553/
29. Oversee a prior art search AI agent with human-in-the-loop by using LangGraph and watsonx.ai - IBM, accessed January 8, 2026, https://www.ibm.com/think/tutorials/human-in-the-loop-ai-agent-langraph-wats

onx-ai

30. Humans in the Loop: The Design of Interactive AI Systems | Stanford HAI, accessed January 8, 2026, https://hai.stanford.edu/news/humans-loop-design-interactive-ai-systems
31. Human-in-the-Loop (HITL) for AI Agents: Patterns and Best Practices - YouTube, accessed January 8, 2026, https://www.youtube.com/watch?v=YCFGjLjNOyw
32. ReportBench: Evaluating Deep Research Agents via Academic Survey Tasks - arXiv, accessed January 8, 2026, https://arxiv.org/html/2508.15804v1
33. DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents, accessed January 8, 2026, https://deepresearch-bench.github.io/
34. A Rigorous Benchmark with Multidimensional Evaluation for Deep Research Agents: From Answers to Reports | by Dixon | Medium, accessed January 8, 2026, https://medium.com/@huguosuo/a-rigorous-benchmark-with-multidimensional-evaluation-for-deep-research-agents-from-answers-to-c0fe2dfb79ec
35. DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents | OpenReview, accessed January 8, 2026, https://openreview.net/forum?id=hQ0K2Hhq7H