

Lab 079: Deep Research Agents ? Multi-Step Knowledge Synthesis

Level: L300 Path: All paths Time: ~90 min ? Cost: Free ? Uses mock research trace data
info "Practice type" This lab analyzes a mock deep-research trace. It teaches the multi-agent research workflow and quality metrics without running a live web research agent.

What You'll Learn

How Deep Research Agents use a multi-agent pipeline for knowledge synthesis The Planner ? Researcher ? Writer ? Reviewer architecture and role responsibilities How citation tracking ensures every claim maps back to a source Analyze a 14-step research trace with agent roles, token usage, and quality scores Identify bottlenecks, token distribution, and quality patterns across the pipeline

Introduction

Deep Research Agents implement a multi-step pipeline for producing well-sourced, comprehensive research reports. Instead of a single LLM generating an entire report, the work is divided across specialized agents:

The Pipeline

[diagram or code sample]

Agent | Role | Key Output | Planner | Decomposes the research question into sub-questions and creates a research plan | Sub-questions, search strategy | Researcher | Executes searches, reads sources, extracts key findings with citations | Findings with source citations | Writer | Synthesizes findings into a coherent, well-structured report | Draft report with inline citations | Reviewer | Reviews the draft for accuracy, completeness, and citation quality | Feedback, quality score, approval/revision |

Citation Tracking

Every claim in the final report must trace back to a source. The pipeline tracks:
sources_cited: Number of unique sources cited in each step quality_score: Agent's self-assessed quality of the output (0.0?1.0)

The Scenario

You are a Research Team Lead evaluating a deep research agent system. You have a 14-step research trace (research_trace.csv) from a completed research run. Your job: analyze the trace to understand agent behavior, token usage, quality patterns, and identify optimization opportunities.

info "Mock Data" This lab uses a mock research trace CSV. The data represents a realistic deep research run with 14 steps across 4 agent roles, including planning, multi-source research, writing, and iterative review.

Prerequisites

Requirement | Why | Python 3.10+ | Run the analysis scripts | pandas library | Data manipulation |

[diagram or code sample]

tip "Quick Start with GitHub Codespaces" Open in GitHub Codespaces

All dependencies are pre-installed in the devcontainer.

? Supporting Files

note "Download these files before starting the lab" Save all files to a lab-079/ folder in your working directory.

File | Description | Download | broken_research.py | Bug-fix exercise (3 bugs + self-tests) | ? Download | research_trace.csv | 14-step research trace with agent roles, tokens, and quality | ? Download |

Step 1: Understand the Trace Format

Each row in the trace represents one step in the research pipeline:

Column | Description | step_id | Sequential step number (1-14) | agent_role | Which agent executed this step: planner, researcher, writer, reviewer | action | What the agent did (e.g., decompose_query, search_sources, write_section) | tokens_used | Number of tokens consumed in this step | sources_cited | Number of sources cited in this step's output | quality_score | Quality assessment of this step's output (0.0-1.0) | duration_sec | Time taken for this step in seconds |

Step 2: Load and Explore the Trace

[diagram or code sample]

Expected output:

[diagram or code sample]

Step 3: Analyze Token Usage by Agent

[diagram or code sample]

[diagram or code sample]

tip "Optimization Insight" The Researcher typically consumes the most tokens because it processes multiple sources per sub-question. To reduce costs, consider caching source extractions and limiting the number of sources per sub-question.

Step 4: Analyze Citation Flow

[diagram or code sample]

Step 5: Quality Analysis

[diagram or code sample]

warning "Quality Variance" Watch for quality drops in later Researcher steps ? this often indicates source exhaustion (the agent is finding lower-quality sources for harder sub-questions). Consider adding a quality threshold that triggers re-search with alternative queries.

Step 6: Build the Research Analysis Report

[diagram or code sample]

? Bug-Fix Exercise

The file lab-079/broken_research.py contains 3 bugs that produce incorrect research analysis. Can you find and fix them all?

Run the self-tests to see which ones fail:

[diagram or code sample]

You should see 3 failed tests. Each test corresponds to one bug:

Test | What it checks | Hint | Test 1 | Total sources cited | Should sum sources_cited, not count rows | Test 2 | Writer token count | Should filter agent_role == "writer", not "researcher" | Test 3 | Researcher step count | Should count rows where agent_role == "researcher", not sum tokens |

Fix all 3 bugs, then re-run. When you see All passed!, you're done!

? Knowledge Check

question "Q1 (Multiple Choice): What is the primary advantage of a multi-agent pipeline over a single-LLM approach for research?"

- A) It uses fewer tokens overall - B) Each agent specializes in one task, enabling better quality and traceability - C) It requires only one model deployment - D) It eliminates the need for citations

??? success "? Reveal Answer" Correct: B) Each agent specializes in one task, enabling better quality and traceability

By splitting research into planning, searching, writing, and reviewing, each agent can be optimized for its specific task. The Researcher can focus on source quality, the Writer on prose coherence, and the Reviewer on factual accuracy. This specialization typically

produces higher-quality output than a single end-to-end generation.

question "Q2 (Multiple Choice): Why is citation tracking important in deep research agents?"

- A) It reduces token usage - B) It ensures every claim maps back to a source, enabling verification and trust - C) It makes the report longer - D) It is required by the LLM's terms of service

??? success "?" Reveal Answer" Correct: B) It ensures every claim maps back to a source, enabling verification and trust

Citation tracking creates an auditable chain from each claim in the final report back to its source. This enables reviewers to verify factual accuracy, users to explore primary sources, and organizations to maintain research integrity ? critical for high-stakes applications like legal, medical, or financial research.

question "Q3 (Run the Lab): What is the total number of sources cited across all steps?"

Run the Step 4 analysis on ? research_trace.csv and sum the sources_cited column.

??? success "?" Reveal Answer" 10 sources

The sum of all sources_cited values across the 14 steps equals 10. Most sources are cited during Researcher steps, with some additional citations added during the Writer's synthesis.

question "Q4 (Run the Lab): How many total tokens did the Writer agent consume?"

Run the Step 3 analysis and find the total tokens for the writer role.

??? success "?" Reveal Answer" Sum of tokens_used where agent_role == "writer"

The Writer's total token count includes all writing and synthesis steps. Filter the trace for agent_role == "writer" and sum the tokens_used column to get the exact value.

question "Q5 (Run the Lab): How many steps did the Researcher agent execute?"

Count the rows where agent_role == "researcher".

??? success "?" Reveal Answer" 6 steps

The Researcher executed 6 steps ? the most of any agent role. This makes sense because the Researcher handles multiple sub-questions from the Planner, with each sub-question potentially requiring multiple search and extraction steps.

Summary

Topic | What You Learned | Deep Research Agents | Multi-agent pipeline for knowledge synthesis with citation tracking | Pipeline Architecture | Planner ? Researcher ? Writer ? Reviewer with specialized roles | Citation Tracking | Every claim maps back to a source across the pipeline | Token Distribution | Researcher uses most tokens; Writer synthesizes; Reviewer validates | Quality Patterns | Quality varies by step ? later research steps may show source exhaustion | Optimization | Cache sources, parallelize research, add quality gates |

Next Steps

Lab 034 ? Multi-Agent with Semantic Kernel (build the agents themselves) Lab 067 ? GraphRAG (enhance research with knowledge graph retrieval) Lab 033 ? Agent Observability (monitor deep research pipelines in production) Lab 076 ? Microsoft Agent Framework (implement pipelines with MAF Graph Workflows)

Phase 3 · Simulation / Real path

Simulation Real path

Preserve the existing free simulation and its fixtures. Estimated cost: US\$0.

Reuse the same acceptance criteria against the documented hosted service. Before running, record a service-specific estimate and a hard spending cap. After validation: Delete hosted deployments, endpoints, indexes, traces, and test data after validation.

Supporting kit

The site build generates stable PDF and ZIP downloads. The ZIP contains the lab guide and

every supporting file declared for this lab.

Download generated PDF Download generated ZIP

Supporting file: broken_research.py

```
/usr/bin/env python3 """Bug-fix: Broken Research Analyzer - Lab 079.""" import io, pandas
as pd def total_sources_cited(trace): return trace["sources_found"].sum() def
writer_token_usage(trace): return trace["tokens_used"].mean() def avg_quality(trace):
return trace["quality_score"].min() def run_tests():
csv="step_id,agent,tokens_used,sources_found,sources_cited,quality_score\nS1,Planner,280,0,0
,0.95\nS2,Researcher,0,12,0,0.90\nS3,Writer,1200,0,5,0.92\nS4,Reviewer,600,0,5,0.88"
t=pd.read_csv(io.StringIO(csv));p,f=0,0 s=total_sources_cited(t);exp=10 if
s==exp:print(f"OK T1: {s}");p+=1 else:print(f"FAIL T1: {s} (exp {exp})");f+=1
w=writer_token_usage(t);exp=1200.0 if abs(w-exp)<1:print(f"OK T2: {w:.0f}");p+=1
else:print(f"FAIL T2: {w:.0f} (exp {exp:.0f})");f+=1 a=avg_quality(t);exp=0.9125 if
abs(a-exp)<0.01:print(f"OK T3: {a:.3f}");p+=1 else:print(f"FAIL T3: {a:.3f} (exp
{exp:.3f})");f+=1 print(f"\n{'All passed!' if f==0 else f'failed'}") if
__name__=="__main__":print("Tests:\n");run_tests()
```

Supporting file: research_trace.csv

```
step_id,agent,action,input_type,output_type,duration_sec,tokens_used,sources_found,sources_c
ited,quality_score S01,Planner,decompose_query,user_question,research_plan,3.2,280,0,0,0.95
S02,Researcher,web_search,search_query_1,raw_results,5.8,0,12,0,0.90
S03,Researcher,web_search,search_query_2,raw_results,4.5,0,8,0,0.88
S04,Researcher,web_search,search_query_3,raw_results,6.2,0,15,0,0.92
S05,Researcher,read_and_extract,url_1,extracted_facts,8.0,450,1,1,0.85
S06,Researcher,read_and_extract,url_2,extracted_facts,7.5,420,1,1,0.88
S07,Researcher,read_and_extract,url_3,extracted_facts,9.2,510,1,1,0.82
S08,Researcher,read_and_extract,url_4,extracted_facts,6.8,380,1,1,0.90
S09,Researcher,read_and_extract,url_5,extracted_facts,7.0,400,1,1,0.87
S10,Writer,synthesize_report,all_facts,draft_report,15.0,1200,0,5,0.92
S11,Reviewer,check_citations,draft_report,review_notes,8.5,600,0,5,0.88
S12,Reviewer,check_accuracy,draft_report,accuracy_report,10.0,700,0,5,0.85
S13,Writer,revise_report,review_notes,final_report,12.0,900,0,5,0.94
S14,Planner,quality_gate,final_report,approval,2.0,150,0,0,0.96
```