SIIM CAIMI25
Conference on Artificial Intelligence in Medical Imaging
OCT 20-21 | UNIVERSITY OF CALIFORNIA, IRVINE

Latterview: A Multi-Agent Framework for Systematic Review Automation using Large Language Models

Pouria Rouzrokh, MD, MPH, MHPE, Yale New Haven Hospital, Yale University

Introduction/Background

Systematic literature reviews are vital for evidence-based medicine but are time-consuming and labor-intensive. We introduce and evaluate an open-source, multi-agent framework utilizing large language models (LLMs) to automate and accelerate key steps in the systematic medical literature review process, aiming to enhance efficiency while maintaining rigor.

Methods/Intervention

We developed LatteReview, a Python framework employing configurable LLM agents (e.g., for title/abstract screening, relevance scoring, data abstraction) orchestrated within customizable multi-round workflows. The framework supports various LLM providers (cloud/local), Retrieval-Augmented Generation, and structured outputs. We evaluated a representative workflow using two 'junior' LLM agents (Gemini-1.5-flash, GPT-4O-mini) for initial title/abstract screening based on inclusion/exclusion criteria, with a 'senior' LLM agent (GPT-4O) resolving disagreements. Evaluation used six diverse public systematic review datasets (SYNERGY collection) and a custom dataset derived from prior cardiothoracic imaging AI reviews, featuring progressively complex criteria. Performance was measured by Area Under the Curve (AUC) and accuracy, evaluated across various decision thresholds.

Results/Outcome

Across the six SYNERGY datasets, the automated workflow demonstrated substantial discriminative capability with AUCs ranging from 0.71 to 0.95. Performance variability was observed, linked to heterogeneity in criteria clarity and dataset inclusion rates (0.9% to 12.4%). On the custom dataset with well-defined criteria, AUCs ranged from 0.79 (most complex criteria) to 0.94 (simplest criteria). Furthermore, by refining agent prompts and calibrating the decision threshold using a 150-record validation set per task, performance was significantly enhanced, achieving accuracy of 99% on subsequent unseen data. Efficiency testing showed processing of 1000 abstracts took approximately 1 minute costing ~$1.20 using the selected models.

Conclusion

LLM-driven multi-agent systems, as implemented in our framework, can effectively automate significant portions of the systematic review process, particularly title and abstract screening. Performance is promising and improves markedly with well-defined review criteria and thoughtful workflow design.

Statement of Impact

Automating systematic reviews using AI tools like LatteReview can drastically accelerate the synthesis of medical research findings. This enables faster development and updating of evidence-based clinical guidelines, quicker assessment of new diagnostic or therapeutic advancements, and helps clinicians and researchers stay current with the rapidly expanding body of literature.

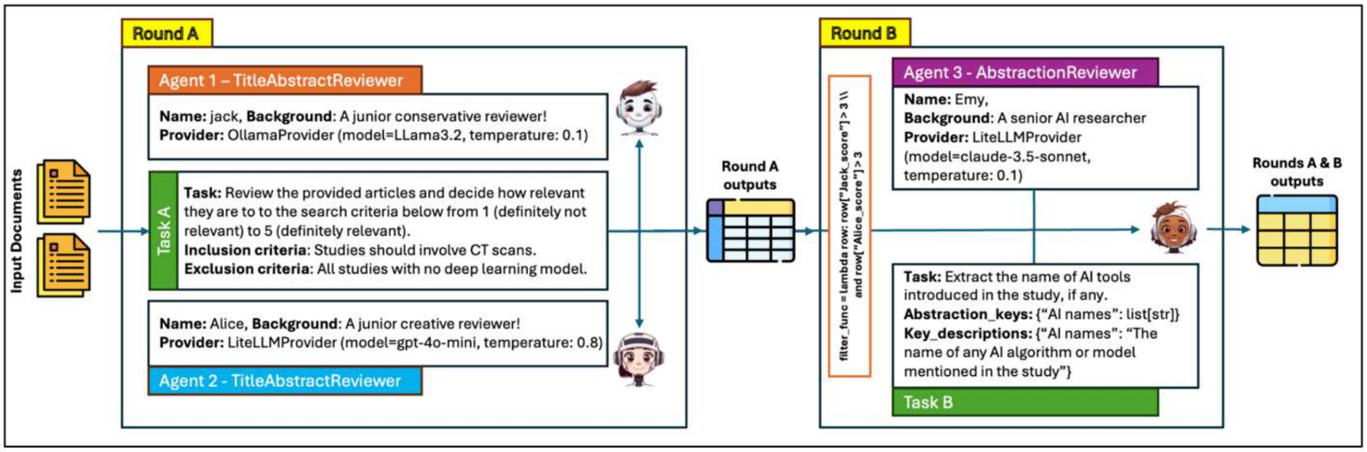
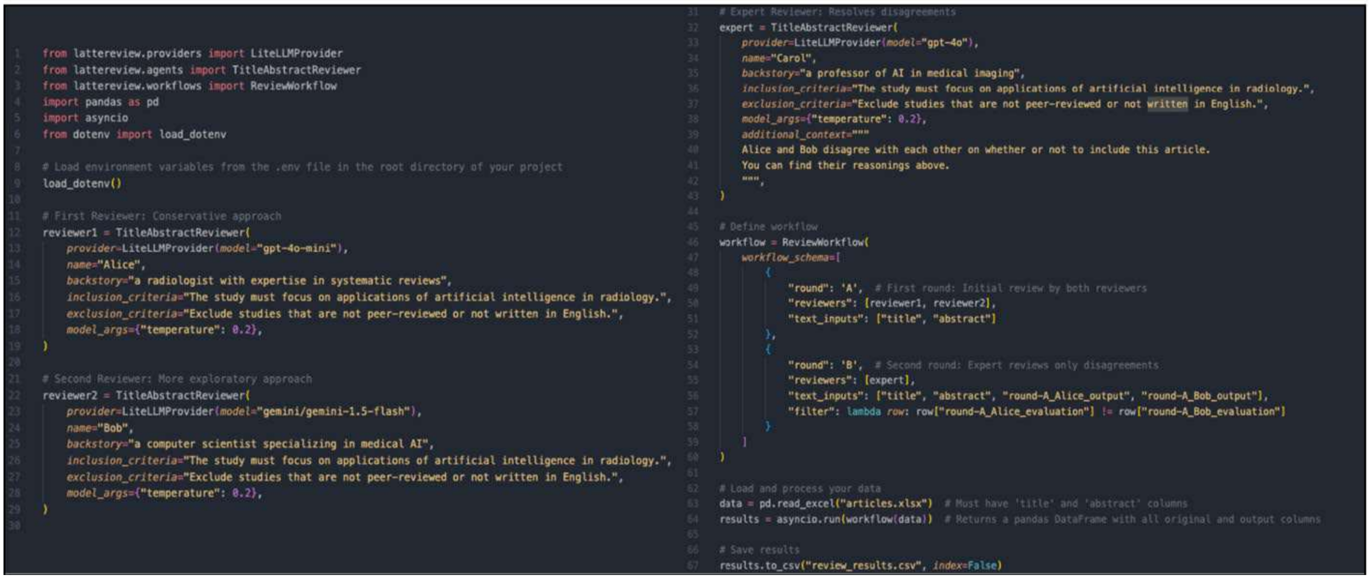
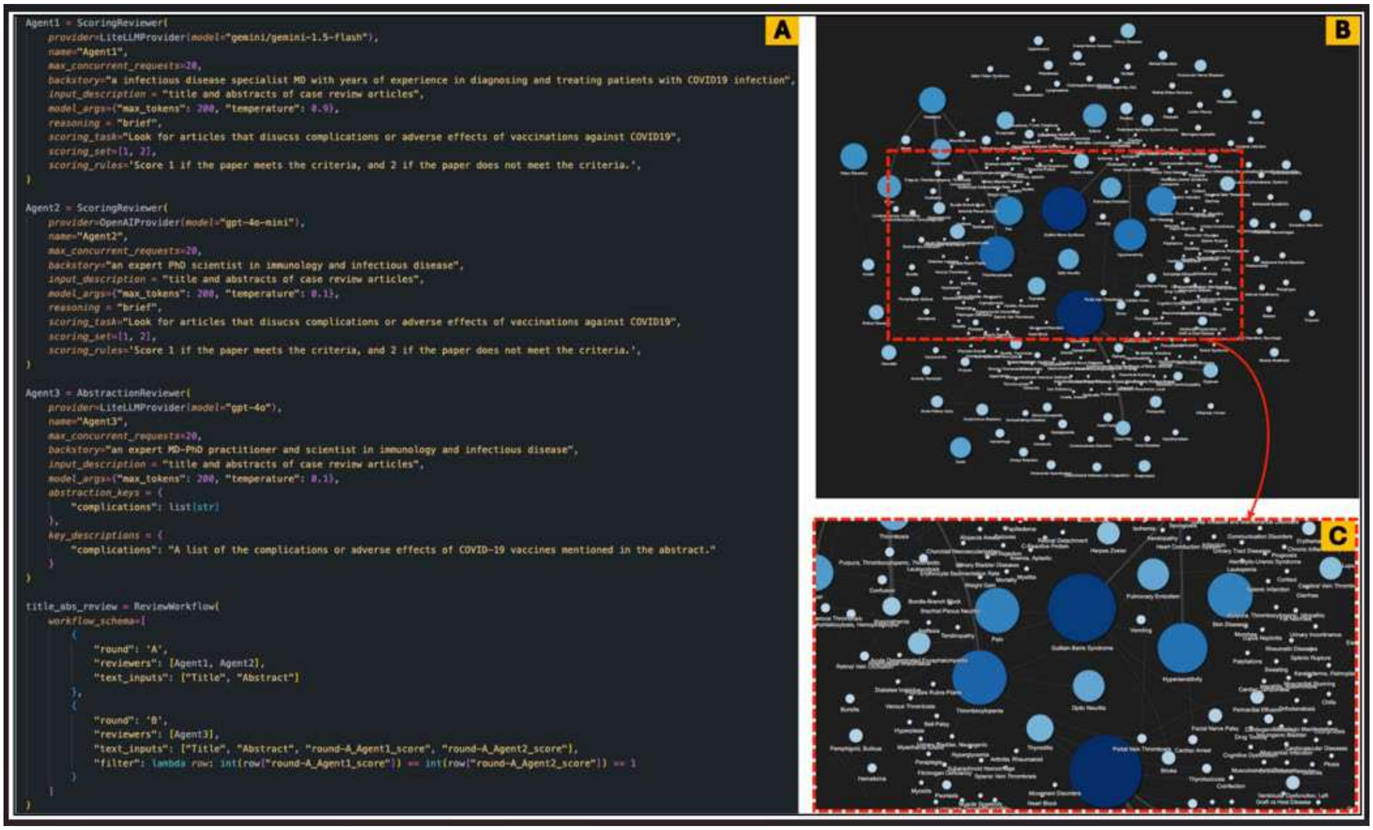

Input Documents
Round A
Agent 1 – TitleAbstractReviewer
Name: jack, Background: A junior conservative reviewer!
Provider: OllamaProvider (model=LLama3.2, temperature: 0.1)
Task A
Task: Review the provided articles and decide how relevant they are to to the search criteria below from 1 (definitely not relevant) to 5 (definitely relevant).
Inclusion criteria: Studies should involve CT scans.
Exclusion criteria: All studies with no deep learning model.
Name: Alice, Background: A junior creative reviewer!
Provider: LiteLLMProvider (model=gpt-4o-mini, temperature: 0.8)
Agent 2 - TitleAbstractReviewer
Round A outputs
Round B
filter_func = lambda row: row["Jack_score"]> 3 \\ and row["Alice_score"]> 3
Agent 3 - AbstractionReviewer
Name: Emy,
Background: A senior AI researcher
Provider: LiteLLMProvider (model=claude-3.5-sonnet, temperature: 0.1)
Task: Extract the name of AI tools introduced in the study, if any.
Abstraction_keys: {"AI names": list[str]}
Key_descriptions: {"AI names": "The name of any AI algorithm or model mentioned in the study"}
Task B
Rounds A & B outputs

Figure 1. An example multiagent review workflow for title and abstract screening with two junior reviewer agents (round A) followed by concept extraction by a senior and more powerful reviewer agent (round B).



```python
from lattereview.providers import LiteLLMProvider
from lattereview.agents import TitleAbstractReviewer
from lattereview.workflows import ReviewWorkflow
import pandas as pd
import asyncio
from dotenv import load_dotenv

# Load environment variables from the .env file in the root directory of your project
load_dotenv()

# First Reviewer: Conservative approach
reviewer1 = TitleAbstractReviewer(
    provider=LiteLLMProvider(model="gpt-4o-mini"),
    name="Alice",
    backstory="a radiologist with expertise in systematic reviews",
    inclusion_criteria="The study must focus on applications of artificial intelligence in radiology.",
    exclusion_criteria="Exclude studies that are not peer-reviewed or not written in English.",
    model_args={"temperature": 0.2},
)

# Second Reviewer: More exploratory approach
reviewer2 = TitleAbstractReviewer(
    provider=LiteLLMProvider(model="gemini/gemini-1.5-flash"),
    name="Bob",
    backstory="a computer scientist specializing in medical AI",
    inclusion_criteria="The study must focus on applications of artificial intelligence in radiology.",
    exclusion_criteria="Exclude studies that are not peer-reviewed or not written in English.",
    model_args={"temperature": 0.2},
)

# Expert Reviewer: Resolves disagreements
expert = TitleAbstractReviewer(
    provider=LiteLLMProvider(model="gpt-4o"),
    name="Carol",
    backstory="a professor of AI in medical imaging",
    inclusion_criteria="The study must focus on applications of artificial intelligence in radiology.",
    exclusion_criteria="Exclude studies that are not peer-reviewed or not written in English.",
    model_args={"temperature": 0.2},
    additional_context="""
    Alice and Bob disagree with each other on whether or not to include this article.
    You can find their reasonings above.
    """,
)

# Define workflow
workflow = ReviewWorkflow(
    workflow_schema=[
        {
            "round": 'A',  # First round: Initial review by both reviewers
            "reviewers": [reviewer1, reviewer2],
            "text_inputs": ["title", "abstract"]
        },
        {
            "round": 'B',  # Second round: Expert reviews only disagreements
            "reviewers": [expert],
            "text_inputs": ["title", "abstract", "round-A_Alice_output", "round-A_Bob_output"],
            "filter": lambda row: row["round-A_Alice_evaluation"] != row["round-A_Bob_evaluation"]
        }
    ]
)

# Load and process your data
data = pd.read_excel("articles.xlsx")  # Must have 'title' and 'abstract' columns
results = asyncio.run(workflow(data))  # Returns a pandas DataFrame with all original and output columns

# Save results
results.to_csv("review_results.csv", index=False)
```

Figure 2. An example workflow for systematic review of some articles using the TitleAbstractReviewer agents.



```
Agent1 = ScoringReviewer(
    provider=LiteLLMProvider(model="gemini/gemini-1.5-flash"),
    name="Agent1",
    max_concurrent_requests=20,
    backstory="a infectious disease specialist MD with years of experience in diagnosing and treating patients with COVID19 infection",
    input_description = "title and abstracts of case review articles",
    model_args={"max_tokens": 200, "temperature": 0.9},
    reasoning = "brief",
    scoring_task="Look for articles that disucss complications or adverse effects of vaccinations against COVID19",
    scoring_set=[1, 2],
    scoring_rules='Score 1 if the paper meets the criteria, and 2 if the paper does not meet the criteria.',
)

Agent2 = ScoringReviewer(
    provider=OpenAIProvider(model="gpt-4o-mini"),
    name="Agent2",
    max_concurrent_requests=20,
    backstory="an expert PhD scientist in immunology and infectious disease",
    input_description = "title and abstracts of case review articles",
    model_args={"max_tokens": 200, "temperature": 0.1},
    reasoning = "brief",
    scoring_task="Look for articles that disucss complications or adverse effects of vaccinations against COVID19",
    scoring_set=[1, 2],
    scoring_rules='Score 1 if the paper meets the criteria, and 2 if the paper does not meet the criteria.',
)

Agent3 = AbstractionReviewer(
    provider=LiteLLMProvider(model="gpt-4o"),
    name="Agent3",
    max_concurrent_requests=20,
    backstory="an expert MD-PhD practitioner and scientist in immunology and infectious disease",
    input_description = "title and abstracts of case review articles",
    model_args={"max_tokens": 200, "temperature": 0.1},
    abstraction_keys = {
        "complications": list[str]
    },
    key_descriptions = {
        "complications": "A list of the complications or adverse effects of COVID-19 vaccines mentioned in the abstract."
    }
)

title_abs_review = ReviewWorkflow(
    workflow_schema=[
        {
            "round": 'A',
            "reviewers": [Agent1, Agent2],
            "text_inputs": ["Title", "Abstract"]
        },
        {
            "round": 'B',
            "reviewers": [Agent3],
            "text_inputs": ["Title", "Abstract", "round-A_Agent1_score", "round-A_Agent2_score"],
            "filter": lambda row: int(row["round-A_Agent1_score"]) == int(row["round-A_Agent2_score"]) == 1
        }
    ]
)
```

Figure 3. (A) A review workflow for screening titles and abstracts from a dataset created by querying PubMed for case report articles on complications of COVID-19 vaccines. The workflow consists of two agents that detect relevant articles and a third agent that abstracts the complications mentioned in those articles into a list. (B and C) A knowledge graph of complications extracted by the reviewers, where each node represents a complication. The size of the nodes correlates with the number of articles mentioning that complication, with larger and darker nodes indicating higher mention counts.

Keywords

Artificial Intelligence; Natural Language Processing; Large Language Models; Systematic Review; Publication Search; Automation