Hatice Nur Kömürcü
310201106
Deniz Karabacak
333078012
CENG568

07.11.2025

Abstract

Generating the "Related Work" section of an academic paper is a complex task that requires synthesis and critical analysis, which Large Language Models (LLMs) often fail to accomplish, producing hallucinations and poor citations. This project aims to design an autonomous Multi-Agent System that overcomes these limitations by simulating the iterative, collaborative workflow of human researchers. Architected using LangGraph for stateful orchestration, the system decomposes the task into a graph of specialized agents, including a Planner, Outliner, Writer, and a crucial Critic agent, which enables a self-correction loop essential for ensuring high-quality, factually grounded output. The entire system is built to run locally on an NVIDIA RTX 4080 Laptop GPU with 12GB VRAM, a constraint managed by serving a 4-bit quantized Qwen3-14B model via the vLLM high-throughput inference engine. The LangChain framework provides the foundational components, while LangSmith delivers end-to-end observability for tracing and debugging the agentic interactions. The objective is to produce thematically coherent and citably accurate "Related Work" sections, with performance validated against datasets, such as SciReviewGen, using evaluation frameworks like SurGE and GREP.

Annotated Bibliography

“SURGE: a benchmark and evaluation framework for scientific survey generation.” Available:

https://arxiv.org/html/2508.15658v2#S1

This paper introduces a comprehensive benchmark and dataset designed to evaluate systems that automatically generate scientific surveys. In our project, which aims to develop a multi-agent system capable of producing related work sections, we plan to use the SurGE benchmark and dataset as both a reference and evaluation foundation. Specifically, the dataset’s structured mapping between scientific papers, extracted key concepts, and summarized survey segments will serve as ground truth for training and assessing our agents’ coordination and synthesis capabilities. Moreover, SurGE’s evaluation metrics will guide the design of our own assessment criteria, allowing us to quantitatively compare our system’s output quality with existing survey generation models. oneal2000/SurGE (github)

“Expert Preference-based Evaluation of Automated Related Work Generation.” Available:

https://arxiv.org/pdf/2508.07955

This paper presents an evaluation framework that incorporates expert judgments to assess the quality of generated related work sections. In our project, we intend to adopt this evaluation approach to complement the quantitative metrics derived from the SurGE benchmark. By

integrating expert preference-based assessment, we aim to capture qualitative dimensions such as coherence, relevance, and scholarly tone that automated metrics may overlook. This framework will help us validate whether our multi-agent system produces related work sections that align with human expert expectations.
UKPLab/arxiv2025-expert-eval-rw: The repository for "Expert Preference-based Evaluation of Automated Related Work Generation" (github)

"LiRA: A Multi-Agent Framework for Reliable and Readable Literature Review Generation."

Available: https://arxiv.org/pdf/2510.05138

This paper introduces a multi-agent system architecture that coordinates specialized agents to produce coherent and trustworthy literature reviews. The framework's modular design and use of vLLM for efficient large language model inference make it particularly relevant to our project. We plan to adopt similar architectural principles and integrate vLLM into our own system to enhance scalability and inference efficiency. By drawing on LiRA's agent coordination strategies and reliability mechanisms, our implementation aims to improve both the factual accuracy and readability of the automatically generated related work sections.
lira-workflow/auto-review-writing: Repository for LiRA: A Multi-Agent Framework for Reliable and Readable Literature Review Generation. (github)

[2305.15186] SciReviewGen: A Large-scale Dataset for Automatic Literature Review Generation (dataset used by LiRA)

"Agent Laboratory: Using LLM Agents as Research Assistants." Available:

https://arxiv.org/pdf/2501.04227

This paper proposes a flexible experimental environment for developing and testing large language model (LLM)-based agents in research workflows. Its modular code architecture and clearly defined agent roles provide a practical foundation for building coordinated multi-agent systems. In our project, we plan to draw inspiration from Agent Laboratory's design principles to structure our agents' interactions and responsibilities more effectively. Specifically, we will adapt its modular framework to organize agent communication, task delegation, and result integration within our related work generation pipeline.

https://youtu.be/DU_W9tgFcqo?si=fbipuHIxXRAaBHeX

Software & Hardware Specifications

This project is designed to operate locally, focusing on performance, cost-effectiveness, and data privacy. The chosen technology stack is intended to function efficiently within the defined hardware constraints.

1. Primary Hardware: NVIDIA GeForce RTX 4080 Laptop GPU

The system will be developed and tested on a laptop equipped with an NVIDIA GeForce RTX 4080 (Laptop) GPU. The critical specification of this hardware is its 12 GB of GDDR6 VRAM. All software strategies are architected to ensure the model can run efficiently within this VRAM budget. Also, we have the option to go with the Nvidia RTX 5070 Laptop GPU since it offers faster 4-bit support with NVFP4 kernels.

2. Language Model: Qwen3-14B (4-bit Quantized)

Given our hardware constraints, a full-precision model is not viable. We will therefore utilize a 4-bit quantized version of a state-of-the-art model.

Our primary candidate is Qwen3-14B. This model is selected for its high performance on reasoning and summarization benchmarks, where even its smaller variants compete with larger models from other families.

Quantization Strategy: We will use a 4-bit GPTQ quantized model or a similar quantization format. A 14B parameter model at 4-bit precision is estimated to require approximately 9-10 GB of VRAM for an 8K context length, which fits within our 12 GB hardware limit.

3. Inference Engine: vLLM

To serve the quantized model, we will use vLLM as our inference engine. This choice is critical for a multi-agent system, which requires numerous and often concurrent LLM calls.

Performance: vLLM provides significantly higher throughput and more efficient memory management than standard Hugging Face or Ollama servers. It achieves this through core innovations like PagedAttention, which optimizes the management of the KV cache.

4. Agentic Framework: LangChain, LangGraph, and LangSmith

The multi-agent system's logic, orchestration, and monitoring will be built using the LangChain ecosystem.

LangChain: Provides the foundational abstractions and components for interfacing with the vLLM server, managing prompts, and defining toolsets for our agents.

LangGraph: This is the core framework for building our multi-agent system. Instead of simple linear chains, LangGraph allows us to define the agentic workflow as a stateful graph. This is essential for enabling complex, cyclical interactions, such as a "Critic" agent evaluating a "Writer" agent's work and routing it back for revisions.

LangSmith: This platform is indispensable for observability. In a complex, non-deterministic multi-agent system, debugging is a primary challenge. LangSmith provides a complete tracing solution to monitor, evaluate, and debug every step of the agents' reasoning and tool-use, ensuring we can pinpoint failures and optimize performance.