

# AUTOMATED SYSTEMATIC REVIEW USING MIXTURE OF AGENTS

*Report by*  
**ASWATHY LOFY RAJ**

In Partial Fulfillment of the Requirements  
for the Degree Of  
**MSc Computer Science with Specialization in Data Analytics**



Supervisors: Dr.Anoop Kadan, Dr.Shamjid P

**School of Digital Sciences**  
**KERALA UNIVERSITY OF DIGITAL SCIENCES, INNOVATION AND  
TECHNOLOGY**

May 2025

# Abstract

Systematic reviews (SRs) are essential for synthesizing scientific evidence, but they are frequently hampered by their time-consuming and labor-intensive nature. This project uses a Mixture-of-Agents (MoA) architecture to provide an automated framework for systematic review generation. The suggested system consists of a collection of specialized, modular agents that work in a coordinated pipeline that replicates the manual SR process. These agents are in charge of literature search, time filtering, title and abstract screening, quality assessment, and review synthesis.

We use cutting-edge large language models (LLMs), such as DeepSeek and Mixtral, to carry out crucial functions like review creation and content screening. APIs from arXiv and Semantic Scholar are used to access external knowledge sources, guaranteeing comprehensive and up-to-date data coverage. In contrast to conventional SR techniques, our method greatly improves reproducibility, transparency, and manual labor.

This study surveys existing SR automation tools, identifies their drawbacks (such as bias, interpretability issues, and limited task generalization), and illustrates how our method overcomes these issues in addition to describing the MoA-based pipeline. The effectiveness of the system is demonstrated through a case study that highlights enhancements in accuracy, efficiency, and adaptability. The suggested framework advances evidence-based research by providing a scalable, transparent, and reliable solution for next-generation SR automation.

# Acknowledgement

I would like to express my sincere gratitude to **Dr. Anoop Kadan, Lecturer in Machine Learning, University of Southampton, United Kingdom**, my project supervisor, for his invaluable guidance, expert insights, and continuous encouragement throughout the course of this work. His mentorship was instrumental in shaping the direction and quality of this research.

I am also profoundly thankful to **Dr. Shamjid P**, my internal guide, for his dedicated support, timely feedback, and thoughtful suggestions that greatly enriched the execution of this project.

Finally, I wish to express deep appreciation to my family and friends for their constant support and motivation, which sustained me during the most challenging phases of this journey.

## Certificate

This is to certify that the thesis /report **Automated Systematic Review Using Mixture Of Agents** submitted by **Aswathy Lofy Raj (Reg. No: 233011)** in partial fulfillment of the requirements for the award of **Name of the Degree** is a bonafide record of the work carried out at **Kerala University of Digital Sciences, Innovation and Technology** under my supervision.

Supervisor

Name of Supervisor

Designation

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgement</b>	<b>iii</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background and Significance of Systematic Reviews . . . . .	1
1.2 Motivation . . . . .	3
1.3 Systematic Review Workflow . . . . .	5
1.4 Scope of the Study . . . . .	5
1.5 Review of Existing Tools . . . . .	6
<b>2 Literature Review</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Advances in AI-Assisted Systematic Reviews . . . . .	8
2.2.1 Integration of AI in Review Workflows . . . . .	8
2.2.2 Enhancing Screening and Classification . . . . .	9
2.3 Emerging Trends and Methodological Shifts . . . . .	9
2.3.1 Transition to Adaptive Models . . . . .	9
2.3.2 Emphasis on Pipeline Modularity . . . . .	9
2.3.3 Focus on Interpretability and Trust . . . . .	9
2.3.4 Adoption of Open-Source Tools . . . . .	10
2.4 Research Gaps and Opportunities . . . . .	10
2.5 Conclusion . . . . .	10

<b>3</b>	<b>Methodology</b>	<b>11</b>
3.1	Phase 1: Initial Model Exploration and Limitations . . . . .	11
3.1.1	Data Preprocessing. . . . .	11
3.1.2	Feature Extraction. . . . .	12
3.1.3	Model Selection and Training. . . . .	12
3.1.4	Implementation Details. . . . .	12
3.1.5	Challenges and Observations. . . . .	12
3.2	Phase 2: Mixture-of-Agents Architecture . . . . .	12
3.2.1	Architecture Diagram and Explanation . . . . .	13
3.2.2	Overview of Agents . . . . .	14
3.3	Flow of Data and System Interaction . . . . .	14
3.3.1	Models Used and Technical Stack . . . . .	15
3.3.2	Evaluation Metrics . . . . .	17
3.3.2.1	Quantitative Metrics . . . . .	17
3.3.2.2	Qualitative Metrics . . . . .	18
3.3.3	Frontend Interface . . . . .	18
<b>4</b>	<b>Results and Discussions</b>	<b>20</b>
4.1	Introduction . . . . .	20
4.2	Results . . . . .	20
4.2.1	Results Comparison . . . . .	20
4.2.2	Qualitative Analysis of Agent Outputs . . . . .	22
4.2.3	Discussion . . . . .	22
4.3	Summary . . . . .	23
<b>5</b>	<b>Conclusion and Future Work</b>	<b>25</b>
5.1	Conclusion . . . . .	25
5.2	Future Work . . . . .	26
	<b>References</b>	<b>28</b>

# List of Figures

1.1	Standard workflow for conducting a systematic review. . . . .	4
3.1	Mixture-of-Agents Architecture, adapted from [11] . . . . .	13
3.2	Sysytematic Review using Mixture of Agents (Systematic work flow) . . . . .	15
3.3	User input interface (index.html) . . . . .	18
3.4	Search progress display (home.html) . . . . .	19
3.5	Final result and review output (results.html) . . . . .	19
3.6	Final result for failed output (results.html) . . . . .	19
4.1	Generated Abstract Screening Results – Phase II . . . . .	22
4.2	Quality Scoring and Filtering by MoA Agents . . . . .	23

# List of Tables

1.1	Comparison of Existing Systematic Review Tools . . . . .	7
4.1	Performance of Baseline Models on HPV Dataset . . . . .	21
4.2	Performance of Baseline Models on PAPD Dataset . . . . .	21
4.3	Advantages of MoA vs. Traditional SR Tools . . . . .	24



# Chapter 1

## Introduction

### 1.1 Background and Significance of Systematic Reviews

Systematic reviews (SRs) have become a cornerstone in evidence-based decision-making across numerous disciplines, including but not limited to healthcare, education, psychology, engineering, and public policy. Unlike traditional literature reviews that are often narrative, selective, and influenced by subjective biases, systematic reviews follow a well-defined, rigorous, and replicable methodology. Their purpose is to comprehensively identify, evaluate, and synthesize all relevant studies on a specific research question or hypothesis.

The significance of systematic reviews is especially pronounced in the health sciences, where clinical practice guidelines and policy decisions depend heavily on accurate and up-to-date evidence. For example, the Cochrane Collaboration—a globally recognized authority in evidence synthesis—has played a crucial role in informing clinical decisions, healthcare policy, and patient care protocols by producing high-quality systematic reviews. These reviews are often regarded as the highest level of evidence in the hierarchy of research designs.

Two foundational frameworks frequently guide the formulation and reporting of systematic reviews: PRISMA and PICO. These frameworks ensure the transparency, consistency, and reproducibility of SRs, which are essential for peer validation and real-world application.

#### The PRISMA Framework

PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) is an evidence-based guideline designed to improve the clarity and completeness of reporting in systematic reviews. Introduced in 2009 and updated in 2020, PRISMA provides a 27-item checklist and a four-phase flow diagram that standardize the reporting of SR processes such as study identification, screening, eligibility, and inclusion.

The PRISMA flow diagram visually maps out the number of records identified through database searches, duplicates removed, studies excluded based on title or abstract screening, full-text assessments, and final inclu-

sions. This transparency is critical for ensuring that the review process is reproducible and free from selection bias. Moreover, PRISMA enhances the review's credibility by making it easier for readers and policymakers to assess the robustness and comprehensiveness of the evidence base.

## The PICO Framework

PICO is an acronym that stands for Population, Intervention, Comparison, and Outcome. It is used to formulate focused and answerable clinical or scientific questions, which in turn inform the systematic search strategy. Each element in the PICO structure plays a critical role in narrowing the scope and increasing the precision of the literature search:

- **Population (P):** Describes the specific group or population targeted by the study (e.g., adults with diabetes).
- **Intervention (I):** Details the primary treatment or condition being investigated (e.g., insulin therapy).
- **Comparison (C):** Identifies the alternative to the intervention, if applicable (e.g., oral medication).
- **Outcome (O):** Specifies the expected result or effect (e.g., blood sugar control, complication rates).

By formalizing the research question, PICO ensures that the systematic review remains focused and methodologically sound. It enables researchers to develop targeted search strings that maximize retrieval of relevant studies while minimizing irrelevant results. The PICO framework is especially important in healthcare research, but it has been effectively adapted in other disciplines like education, technology adoption, and social science.

## Challenges in Manual Systematic Reviews

Despite their methodological robustness and value, manual SRs are time-consuming, labor-intensive, and susceptible to human error. Conducting a high-quality review involves multiple steps including formulating a review question, developing inclusion and exclusion criteria, performing comprehensive literature searches, screening titles and abstracts, retrieving and assessing full texts, evaluating methodological quality, extracting data, and synthesizing findings.

The sheer volume of scientific publications being produced further compounds these challenges. For instance, databases like PubMed, arXiv, and Semantic Scholar collectively index tens of millions of articles, with thousands of new entries added monthly. This deluge of information makes it practically infeasible for researchers to manually sift through all potentially relevant studies. Consequently, many reviews either become outdated quickly or are narrowly scoped, potentially omitting critical data.

Moreover, traditional SRs often take 6 to 24 months to complete. This delay can be detrimental in fields where timely evidence is crucial—such as during health emergencies, technological innovation cycles, or policy decision windows. The extended duration also hampers real-time evidence translation, reducing the societal and scientific utility of the review.

## 1.2 Motivation

The pressing need for faster, scalable, and more accurate evidence synthesis has motivated the exploration of AI-driven approaches to automate the systematic review pipeline. Although some tools exist to aid in specific tasks—such as screening or data extraction—few offer integrated, intelligent, and customizable end-to-end solutions.

While platforms like Rayyan, Covidence, and DistillerSR have made important contributions, they tend to focus on isolated components of the SR process and often require considerable manual oversight. Many are also subscription-based and not accessible to researchers in resource-constrained environments. Furthermore, these tools generally lack explainability and are not easily adaptable across research domains.

The motivation for this project arises from this gap: the need for a modular, transparent, and domain-agnostic solution that automates multiple phases of systematic review using advanced AI agents. The goal is to build a framework that not only reduces the time and effort required to conduct systematic reviews but also improves consistency, reproducibility, and interpretability.

### 1.2.1 Objectives

- To design and implement an AI-driven framework for automating systematic reviews using a Mixture-of-Agents (MoA) architecture.
- To evaluate traditional machine learning methods and identify their limitations in title and abstract screening tasks.
- To develop modular agents that replicate the core steps of systematic reviews, including literature search, screening, quality assessment, and synthesis.
- To leverage large language models (LLMs) such as DeepSeek and Mixtral for semantic relevance classification and review generation.
- To build an accessible user interface that allows researchers to interact with the system without programming expertise.
- To ensure transparency, modularity, and reproducibility across all stages of the systematic review pipeline.

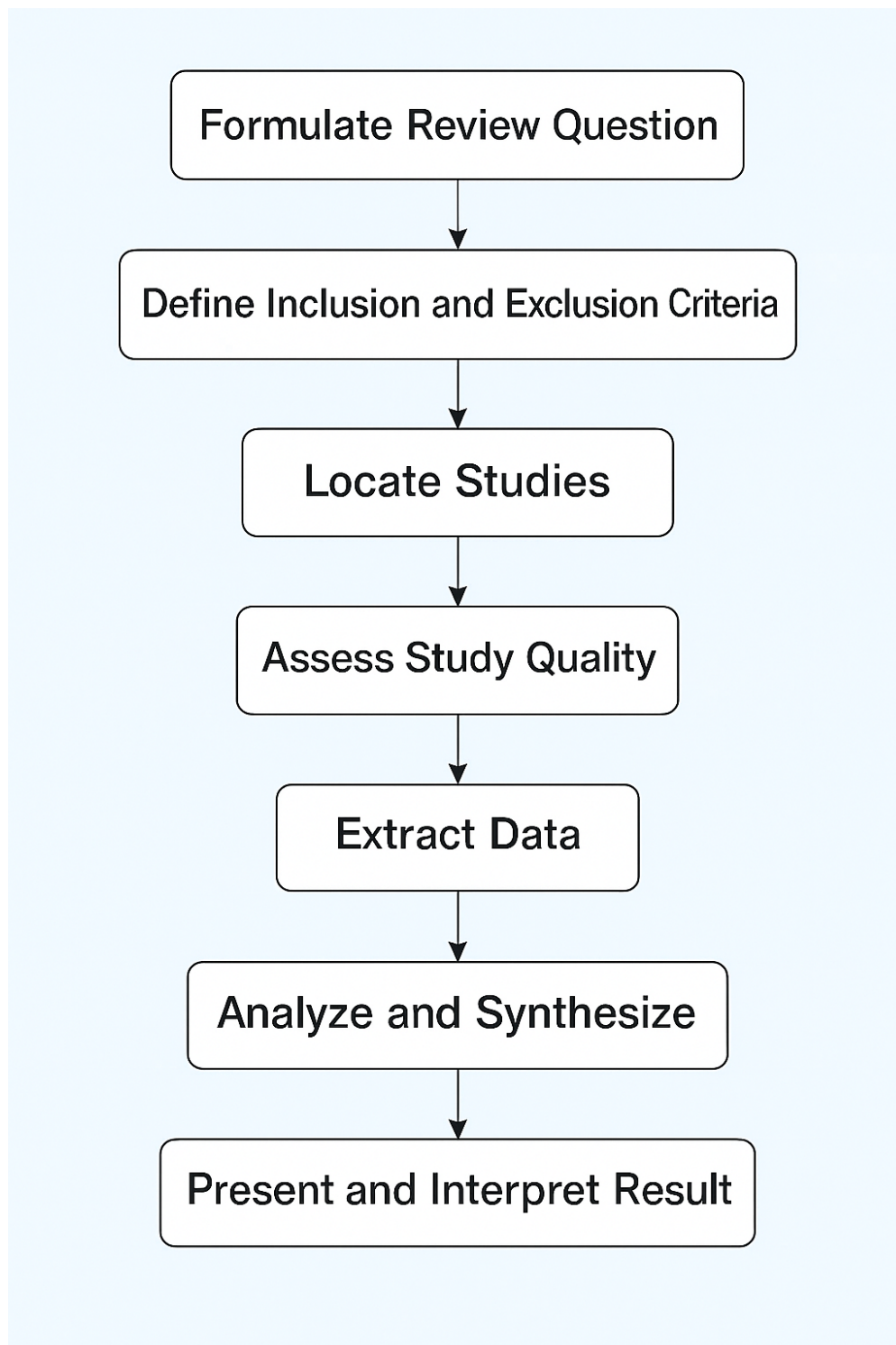


Figure 1.1: Standard workflow for conducting a systematic review.

## 1.3 Systematic Review Workflow

A systematic review typically unfolds through a series of methodologically rigorous steps. The process begins with the formulation of a clearly defined research question, usually derived using frameworks such as PICO. This step ensures that the review has a well-scoped focus and avoids ambiguity in downstream stages.

Following the question formulation, researchers establish explicit inclusion and exclusion criteria. These criteria determine which studies are eligible for review based on aspects such as publication year, study design, population characteristics, interventions, and outcomes. This filtering mechanism is essential for maintaining consistency and objectivity throughout the review process.

Next, a comprehensive literature search is performed across multiple databases like PubMed, Scopus, Web of Science, arXiv, and Semantic Scholar. The aim is to capture all potentially relevant studies, minimizing publication bias and ensuring completeness. The search is typically guided by boolean queries derived from the PICO components.

Once the initial corpus of articles is gathered, the screening phase begins. This involves reviewing the titles and abstracts of the retrieved studies to determine their relevance. Studies that pass this phase proceed to full-text retrieval, where a more detailed eligibility assessment is conducted based on the inclusion and exclusion criteria.

Subsequently, the quality of each eligible study is critically appraised using standardized tools or checklists. This appraisal evaluates aspects such as study design, sample size, data collection methods, and potential sources of bias. High-quality studies are retained for data extraction, where relevant information—such as interventions, outcomes, and statistical measures—is systematically recorded.

The extracted data are then synthesized. Depending on the nature of the data, synthesis may be qualitative (e.g., thematic analysis) or quantitative (e.g., meta-analysis). The goal is to combine the findings in a meaningful way that answers the original research question.

Finally, the results are reported according to established guidelines like PRISMA. This includes detailing the search strategy, selection process, quality assessment, and synthesized outcomes. Transparency at this stage enhances the credibility, reproducibility, and practical utility of the review.

## 1.4 Scope of the Study

This study is focused on the design, development, and evaluation of an AI-driven framework that automates key phases of the systematic review process through a Mixture-of-Agents (MoA) architecture. The solution aims to address both technical and practical challenges associated with manual reviews by incorporating modular, intelligent agents with domain-agnostic capabilities.

The system is designed to retrieve academic literature from open-access databases such as arXiv and Se-

semantic Scholar through publicly available APIs. It then performs intelligent title and abstract screening using large language models (LLMs), which are guided by prompts derived from user-defined inclusion criteria. A quality assessment agent assigns relevance scores to each candidate study using a combination of rule-based filters and semantic reasoning.

In the final stage, a review agent synthesizes the most relevant studies into a cohesive narrative that aligns with academic reporting standards. The output includes both structured data (e.g., labeled articles with scores) and an automatically generated literature review.

Though the prototype is evaluated in the context of computer science and data analytics, the framework is inherently flexible and can be adapted to other domains—such as medicine, education, or economics—with minimal customization. This generalizability, along with the use of open-source models and APIs, ensures that the system is accessible, transparent, and reproducible for a broad community of researchers.

## 1.5 Review of Existing Tools

Over the years, several software tools have emerged to assist researchers in various stages of the systematic review process. While these tools offer useful functionalities—such as citation management, screening support, and collaboration features—they often fall short of providing full automation, domain generality, or open accessibility. Most existing platforms focus on streamlining specific tasks within the SR pipeline rather than offering a fully integrated, intelligent system.

Table 1.1 provides a comparative overview of some widely used systematic review tools.

As shown in the table, while each of these tools contributes to improving review efficiency, none of them offers a fully modular, explainable, and domain-independent solution. These gaps justify the development of a new AI-driven framework that leverages a Mixture-of-Agents (MoA) architecture to deliver integrated, flexible, and open-source automation.

Table 1.1: Comparison of Existing Systematic Review Tools

Tool Name	Automation Capabilities	Open Access	Limitations
Rayyan	Semi-automated screening (title/abstract)	Free with limitations	No full pipeline automation; limited explainability
Covidence	Workflow management, screening, data extraction	Subscription required	Not fully automated; domain-specific tuning required
DistillerSR	Comprehensive review management with AI screening	Commercial (paid license)	Expensive; lacks transparency in algorithmic decisions
ASReview	Active learning for screening	Open source	Only supports screening; not end-to-end
RobotReviewer	Automated risk-of-bias assessment	Open source	Only applies to certain study types; narrow focus

## Chapter 2

# Literature Review

### 2.1 Introduction

Systematic reviews are foundational to evidence-based research, offering comprehensive syntheses of existing literature across various disciplines. Traditionally, these reviews involve meticulous processes, including manual searching, screening, relevance classification, quality assessment, and synthesis. However, these methods are often labor-intensive, time-consuming, and susceptible to human bias. The advent of Artificial Intelligence (AI), particularly Machine Learning (ML) and Natural Language Processing (NLP), has introduced transformative approaches to streamline and enhance the systematic review process.

### 2.2 Advances in AI-Assisted Systematic Reviews

#### 2.2.1 Integration of AI in Review Workflows

Recent studies have explored the integration of AI to automate various stages of systematic reviews. Fabiano et al. [1] provide a comprehensive survey on the utilization of Large Language Models (LLMs) in scientific research, highlighting their potential in automating literature reviews and data extraction processes. Similarly, Ibrahim et al. [2] propose a novel question-answering framework employing LLMs for automated citation screening, demonstrating improved efficiency and accuracy in identifying relevant studies.

Marshall et al. [3] introduce the ASReview LAB platform, which leverages active learning to reduce reviewer workload during the screening phase without compromising recall. Their findings underscore the value of adaptive models that learn from human decisions in real-time, enhancing the efficiency of the review process.



## 2.2.2 Enhancing Screening and Classification

The LitLLMs project[4] benchmarks the performance of state-of-the-art LLMs, such as GPT-4 and Claude, on title and abstract screening tasks across multiple domains. The study reveals that LLMs, when effectively prompted or fine-tuned on domain-specific corpora, can achieve near-human accuracy in relevance classification.

Zhang et al.[5] explore the transformation of screening into an interactive question-answering process powered by BERT-based QA systems. This approach reframes relevance detection through structured PICO (Population, Intervention, Comparator, Outcome) queries, facilitating more precise and context-aware screening.

Elmagarmid et al.[6] emphasize the importance of human-in-the-loop (HITL) architectures, particularly in high-stakes contexts such as pandemic surveillance. By combining transformer models with expert feedback loops, their approach achieves high interpretability and responsiveness, introducing metrics for evaluating human-AI synergy, focusing on decision confidence and reviewer workload.

## 2.3 Emerging Trends and Methodological Shifts

### 2.3.1 Transition to Adaptive Models

There is a noticeable shift from deterministic, rules-based systems to probabilistic and adaptive models capable of learning from interactions. This transition is exemplified by the use of active learning (e.g., ASReview[3]) and prompt engineering for few-shot and zero-shot classification (e.g., LitLLMs[4]).

### 2.3.2 Emphasis on Pipeline Modularity

Recent systems prioritize flexible integration, allowing researchers to selectively automate specific review components. This modularity supports domain customization and ethical compliance, particularly where manual oversight is legally or procedurally mandated.

### 2.3.3 Focus on Interpretability and Trust

Interpretability and trust have gained prominence as critical design goals. Opaque “black-box” systems pose risks in domains where exclusions or inclusions may affect patient care or public policy. The incorporation of explainability mechanisms—such as scoring rationales, feedback interfaces, and hybrid QA structures—marks a shift toward more accountable AI design in evidence synthesis[6],[9].

### 2.3.4 Adoption of Open-Source Tools

An open-source ethos is another recurring theme. Tools like ASReview[3] are community-driven, reproducible, and extensible, enabling continuous improvement and wide-scale adoption. This contrasts sharply with commercial tools that often lack transparency and hinder external validation.

## 2.4 Research Gaps and Opportunities

Despite promising advances, several research gaps persist, pointing to fertile ground for continued innovation:

- **Incomplete End-to-End Automation:** Few systems offer seamless integration across all stages of the review process. A unified architecture leveraging specialized agents for modular tasks remains underexplored[7].
- **Limited Multi-Agent Collaboration:** Most tools focus on isolated tasks. The idea of orchestrating a “mixture of agents”—each tailored to a specific review phase—is still nascent[7].
- **Lack of Real-Time Adaptivity:** Dynamic updates based on feedback or new publications are rare. Reinforcement learning and user-guided pipelines are essential for responsive reviews[6].
- **Contextual Comprehension and Prompt Sensitivity:** LLMs suffer from hallucinations, prompt instability, and domain misalignment. Robust prompting strategies are crucial[2],[4].
- **Insufficient Evaluation Metrics:** Beyond recall and precision, broader metrics like trust, ethical compliance, and reproducibility need emphasis[1],[8].

## 2.5 Conclusion

The literature reviewed highlights an accelerating trajectory in the use of AI to support and enhance systematic reviews. From adaptive active learning platforms to powerful LLM-based classification engines, the landscape is rich with innovations that challenge traditional paradigms. Yet, no existing approach offers a fully integrated, adaptable, and interpretable pipeline that spans the complete review lifecycle.

The gaps in modular integration, collaborative intelligence, and human-AI co-adaptation present both a challenge and an opportunity. By responding to these limitations, future research endeavors can advance the state of the art through multi-agent systems that align technological capability with methodological rigor and practical usability.

## Chapter 3

# Methodology

The methodology adopted for this study is divided into two major phases: an initial phase of experimental model exploration using machine learning and transformer-based classifiers, followed by a modular agent-based design using a Mixture-of-Agents (MoA) framework. The transition from monolithic approaches to a multi-agent pipeline was guided by the limitations of existing systems and the need for a more interpretable, customizable, and interactive automation of the systematic review (SR) process.

### 3.1 Phase 1: Initial Model Exploration and Limitations

The first phase of the project was dedicated to replicating and validating baseline approaches to automate the title and abstract screening task in systematic literature reviews (SLRs), based on the methodology presented by Du et al. [12]. To this end, two manually curated corpora—the HPV corpus and the PAPD corpus—were employed, both of which contain labeled citation metadata including title, abstract, MeSH terms, authors, journal, keywords, and publication types. Each entry was labeled as either “Include” or “Exclude” for systematic review inclusion.

#### 3.1.1 Data Preprocessing.

The raw data was preprocessed to normalize textual inputs across the datasets. This involved lowercasing all text, removing special characters, and concatenating multiple textual fields to generate two feature sets: a baseline set (title and abstract) and an extended set (including MeSH terms, authors, journal, keywords, and publication type). Missing fields were filled with whitespace to ensure structural consistency.

### 3.1.2 Feature Extraction.

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was applied to both baseline and extended feature sets using `TfidfVectorizer` from the `scikit-learn` library with English stopwords removal. This process converted text into sparse numerical vectors suitable for traditional machine learning algorithms.

### 3.1.3 Model Selection and Training.

Four conventional machine learning algorithms—Logistic Regression, Support Vector Machines (SVM), Random Forest, and XGBoost—were implemented. Each model was trained on both the HPV and PAPD datasets using an 8:1:1 split for training, validation, and test sets. Hyperparameters were adopted from the configurations reported by Du et al. [12]. All classifiers were evaluated on both the baseline and extended feature sets.

### 3.1.4 Implementation Details.

The models were implemented in Python using libraries such as `scikit-learn`, `XGBoost`, `pandas`, and `jobjlib`. During training, model performance metrics and serialized model files were automatically saved to structured directories, enabling reproducibility and facilitating future development.

### 3.1.5 Challenges and Observations.

While implementing this phase, several limitations were identified. Traditional ML models, though computationally efficient and interpretable, offered limited capacity to capture deep semantic relationships in text. Although transformer-based models were noted to yield superior performance in related studies [12], they were excluded from this phase due to high resource demands and limited interpretability. Furthermore, incorporating domain-specific logic or enabling intermediate transparency proved difficult with standard end-to-end pipelines. These observations motivated the transition to a modular, agent-based architecture in the next phase of the project.

## 3.2 Phase 2: Mixture-of-Agents Architecture

In the second and final phase, I designed and implemented a Mixture-of-Agents (MoA) pipeline tailored to the systematic review (SR) process. The system was decomposed into specialized agents, each responsible for a distinct task that mimics the sequential stages of a human-conducted SR. Each agent operates autonomously, consuming inputs and producing outputs that are passed to subsequent agents. This architecture offers modularity, extensibility, and transparency, allowing for better user control and iterative refinement.

The Mixture-of-Agents paradigm is inspired by cognitive models of human decision-making, where distinct cognitive functions such as search, judgment, and synthesis are modularized into discrete, interacting units. By simulating this paradigm computationally, I enable each agent to encapsulate a specific logic-driven or language-based operation. This division of labor ensures that the overall workflow is interpretable and adaptable. Each agent is designed with focused responsibilities and scoped knowledge, enabling isolated testing, improvement, or replacement as needed—thereby enhancing maintainability and extensibility of the overall system.

Unlike end-to-end monolithic models, which often obscure the reasoning chain, our MoA design allows inspection at each intermediate stage—title screening, abstract decision, and quality scoring—giving users an opportunity to verify, modify, or rerun any part of the pipeline. This also allows asynchronous execution and independent logging of decisions, which is essential in SR pipelines where traceability is a requirement (e.g., in PRISMA documentation).

### 3.2.1 Architecture Diagram and Explanation

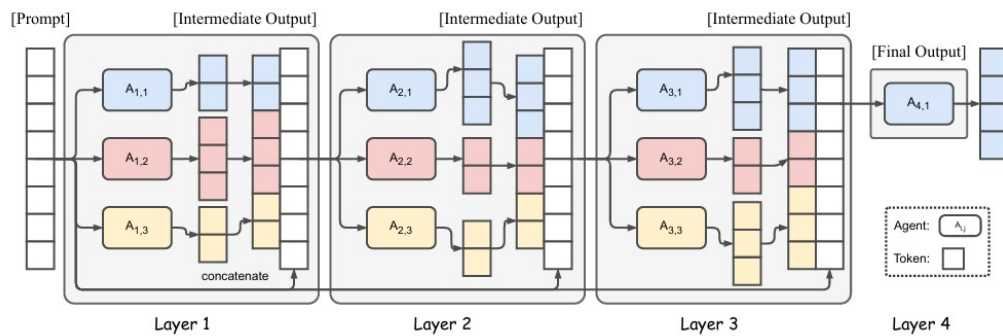


Figure 3.1: Mixture-of-Agents Architecture, adapted from [11]

The Mixture-of-Agents (MoA) architecture is designed to leverage the collaborative strengths of multiple agents—each potentially powered by a distinct large language model (LLM) or specialized logic module. As illustrated in Figure 3.1, the architecture is composed of multiple layers. In each layer, several agents independently process inputs and generate candidate outputs. These outputs are then passed as context to the agents in the subsequent layer, where they are refined or synthesized into higher-quality responses.

Agents are assigned specialized roles: *proposers* generate diverse and informative outputs, while *aggregators* combine and evaluate these to produce coherent summaries or decisions. This collaborative interaction harnesses the phenomenon of *LLM collaborativeness*, whereby models improve their outputs when exposed to peer responses—even if some peers are individually weaker [11].

The MoA framework eliminates the need for fine-tuning and instead operates using prompt-based interfaces, which improves efficiency and scalability. This makes it well-suited for complex multi-step tasks like systematic reviews, where flexibility, transparency, and modularity are crucial.

### 3.2.2 Overview of Agents

The system is composed of the following agents:

1. **Search Agent:** This agent uses public APIs from repositories such as arXiv and Semantic Scholar to retrieve scholarly articles matching the user-specified keywords. The retrieved data includes titles, abstracts, publication years, authors, and URLs.
2. **Time Filter Agent:** This agent filters the retrieved results based on the minimum year specified by the user. It ensures temporal relevance and narrows down the scope of analysis to recent literature.
3. **Title Screening Agent:** A lightweight natural language model (DeepSeek) is queried with the title of each paper along with the inclusion and exclusion criteria. The model outputs a binary decision indicating whether the paper is likely relevant. This simulates the title screening step conducted in manual SRs.
4. **Abstract Screening Agent:** Abstracts of papers that passed the title screening are further evaluated by querying the DeepSeek model with a similar prompt. The screening here is more detailed and forms the core filtration layer.
5. **Quality Agent:** For papers that pass the abstract screening stage, this agent assesses quality based on clarity, structure, and information density. It assigns a numerical score and provides justification. The scoring is designed to help rank papers for final inclusion.
6. **Review Agent:** Using a summarization-capable language model (Mixtral), this agent synthesizes the abstract content of the final set of papers and generates a preliminary literature review. The output is coherent, thematically grouped, and formatted to align with academic standards.

## 3.3 Flow of Data and System Interaction

The MoA system operates in a sequential pipeline, beginning with user input and culminating in a structured literature review. The interaction between agents can be represented as:

The architecture begins with the **User Input**, which serves as the entry point for the user's review topic, inclusion/exclusion criteria, and other parameters. This data flows to the **Search Agent**, which dispatches queries to scholarly repositories. Fetched papers are filtered temporally by the **Time Filter Agent**, and then routed to the **Title Screening Agent**, which applies LLM-based binary classification on the titles. Papers that

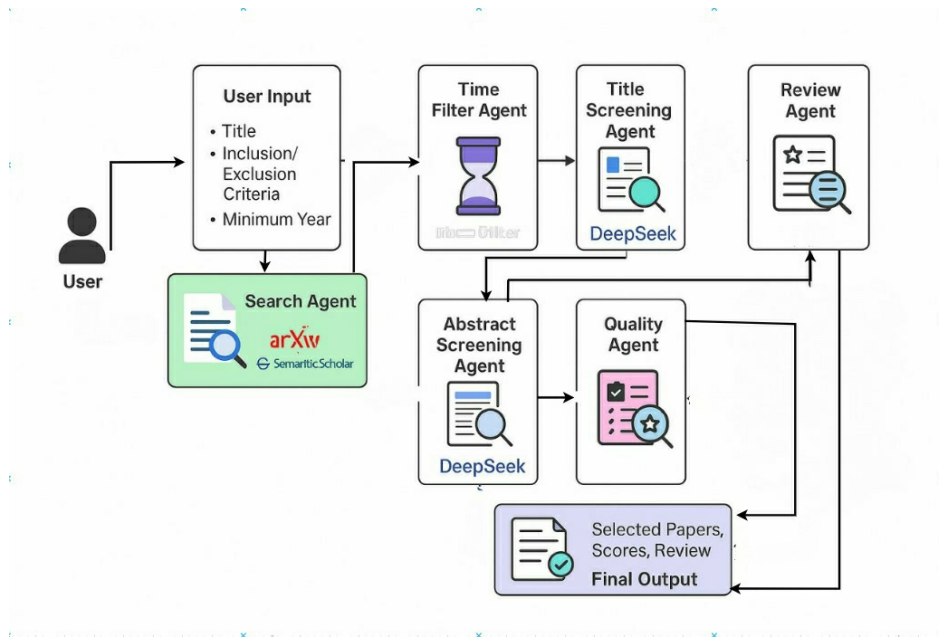


Figure 3.2: Sysytematic Review using Mixture of Agents (Systematic work flow)

pass are forwarded to the **Abstract Screening Agent** for deeper semantic filtering using abstract content. The refined set is then scored by the **Quality Agent**, which appraises readability, structure, and informativeness. Finally, the **Review Agent** synthesizes the retained papers into a draft literature review. This unidirectional agent-to-agent data movement is coordinated by a lightweight orchestration engine, implemented in Python, ensuring consistent execution and logging.

Each agent in the figure is represented as a self-contained module with clearly defined inputs and outputs. The modular arrows illustrate data handoffs, while the feedback loops (for instance, between screening and quality assessment) indicate optional re-screening based on quality thresholds or user overrides. This diagram also demonstrates the MoA's ability to isolate logic into independently deployable services—paving the way for future enhancements like parallel agent execution or user-in-the-loop configurations.

### 3.3.1 Models Used and Technical Stack

The implementation of the proposed system leverages state-of-the-art language models alongside a robust technical infrastructure to enable efficient semantic screening and literature review generation. This section outlines the models employed and the technologies underpinning the backend architecture.

## Models Used

**DeepSeek-LLM:** This model is utilized for semantic screening of academic papers. Given an abstract and a predefined set of inclusion and exclusion criteria, the model evaluates the relevance of the paper. It provides binary classifications (“YES” or “NO”) accompanied by brief justifications, aiding in transparent decision-making during the screening phase.

**Mixtral:** Employed in the review synthesis stage, Mixtral generates comprehensive summaries by integrating insights from multiple selected papers. The model identifies recurring themes, innovative contributions, and knowledge gaps, ultimately producing structured literature reviews suitable for academic dissemination.

## Technical Stack

**Programming Language:** The backend is fully implemented in Python, selected for its readability, community support, and availability of powerful libraries for machine learning and web development.

## API Interactions

- **requests:** Facilitates synchronous HTTP requests to interact with language model APIs hosted initially on OpenRouter.
- **httpx:** Used for making asynchronous HTTP requests to language model APIs. It provides a modern, high-performance interface for sending concurrent requests, improving efficiency in agent communication.

## Data Processing and Scoring:

- **scikit-learn:** Used for implementing preprocessing routines and scoring mechanisms. Papers are evaluated on structural and linguistic parameters, including abstract length, clarity, and presence of scientific components.

**Custom API Clients:** To ensure modularity and abstraction from specific model providers, custom wrappers were developed. These clients manage request formatting, error handling, and response parsing, thus decoupling model logic from API-specific dependencies.

**Environment Management:** The `python-dotenv` package is employed for secure and configurable handling of environment variables, including API keys and endpoint URLs.

**Deployment:** Although the system was initially deployed using OpenRouter for LLM access, ongoing efforts aim to migrate to open-source alternatives hosted on Hugging Face. This transition enhances accessibility, reduces reliance on proprietary APIs, and promotes cost-effective scalability and reproducibility across different environments.



## Conclusion

By integrating DeepSeek-LLM and Mixtral with a modular Python backend and well-defined API infrastructure, the system ensures accurate semantic screening and high-quality literature review generation. The current architecture supports flexibility, extensibility, and reproducibility, thereby laying a strong foundation for future enhancements in AI-assisted systematic review tools.

### 3.3.2 Evaluation Metrics

To evaluate the performance of the screening agents, I employ standard binary classification metrics that help quantify the quality of the system's predictions. These metrics are crucial for assessing the effectiveness of each agent in the systematic review process. The evaluation is based on both quantitative measures (precision, recall, F1 score, and accuracy) and qualitative assessments of the user experience.

#### 3.3.2.1 Quantitative Metrics

- **Precision (P):** Precision measures the proportion of the papers selected as relevant by the agent that are truly relevant. It is defined as:

$$P = \frac{TP}{TP + FP}$$

where  $TP$  is the number of true positives (relevant papers correctly identified), and  $FP$  is the number of false positives (irrelevant papers incorrectly identified as relevant).

- **Recall (R):** Recall assesses the proportion of all relevant papers that are successfully included by the agent. It is given by:

$$R = \frac{TP}{TP + FN}$$

where  $FN$  is the number of false negatives (relevant papers missed by the agent).

- **F1 Score:** The F1 score provides a balance between precision and recall by computing their harmonic mean. It is particularly useful when there is a class imbalance (e.g., when relevant papers are fewer than irrelevant ones). The F1 score is calculated as:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

- **Accuracy:** Accuracy measures the overall proportion of correct predictions made by the agent. It includes both true positives and true negatives (irrelevant papers correctly excluded). It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where  $TN$  represents the number of true negatives.

### 3.3.2.2 Qualitative Metrics

In addition to the quantitative metrics, user satisfaction is evaluated qualitatively by considering two main factors:

- **Interpretability:** Users manually assess the clarity and transparency of the intermediate outputs generated by the screening agents. This evaluation focuses on how easily users can understand the decisions made by the system, especially the classification of papers as relevant or irrelevant.
- **Usability of the Generated Review:** The quality and usefulness of the generated literature review are also critical. This includes evaluating the cohesiveness, depth, and relevance of the review in relation to the user's research goals. Feedback from users regarding the structure and actionable insights provided by the review helps in refining the system's output.

These evaluation metrics, both quantitative and qualitative, offer a comprehensive assessment of the system's performance and ensure that it meets the desired standards for accuracy, interpretability, and user satisfaction.

### 3.3.3 Frontend Interface

To improve accessibility, a minimal Ib-based front end was developed using HTML and JavaScript. The interface allows users to specify their review criteria through a form on `index.html`, view search progress on `home.html`, and see final results and the generated literature review on `results.html`. Screenshots of each page are shown below.

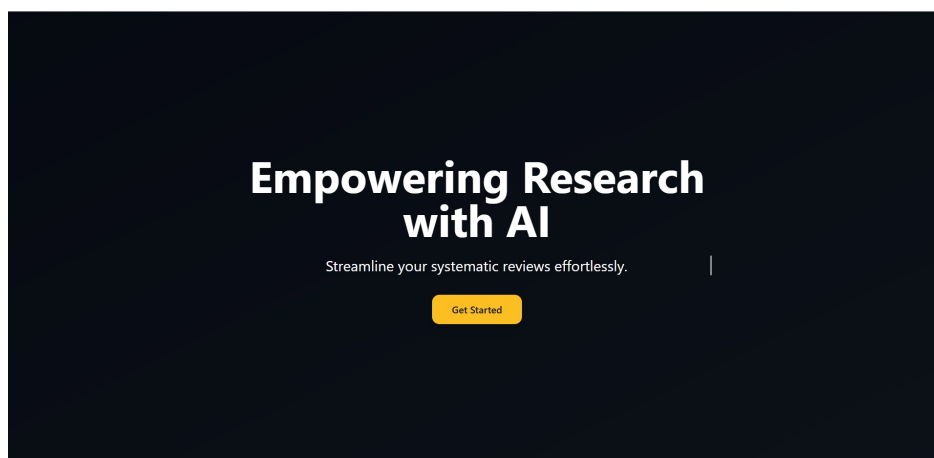


Figure 3.3: User input interface (`index.html`)

This frontend serves as a lightweight gateway for interacting with the MoA system, making the tool usable for researchers with limited programming background.

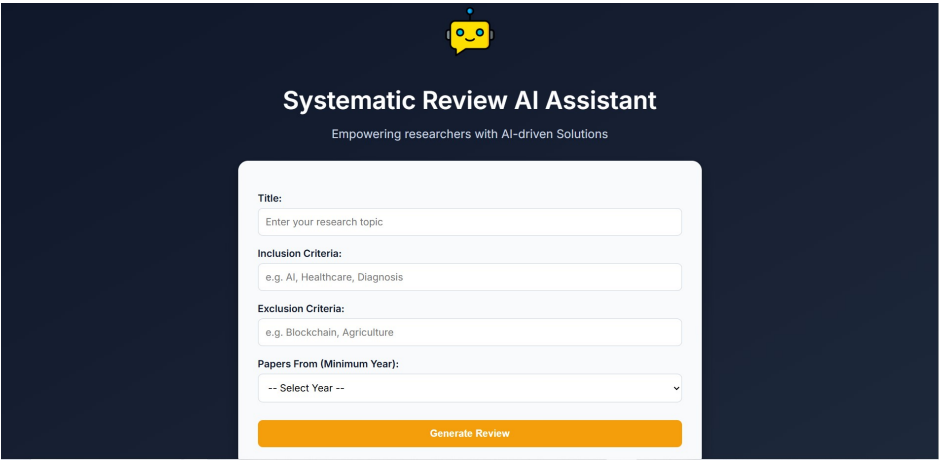


Figure 3.4: Search progress display (home.html)



Figure 3.5: Final result and review output (results.html)

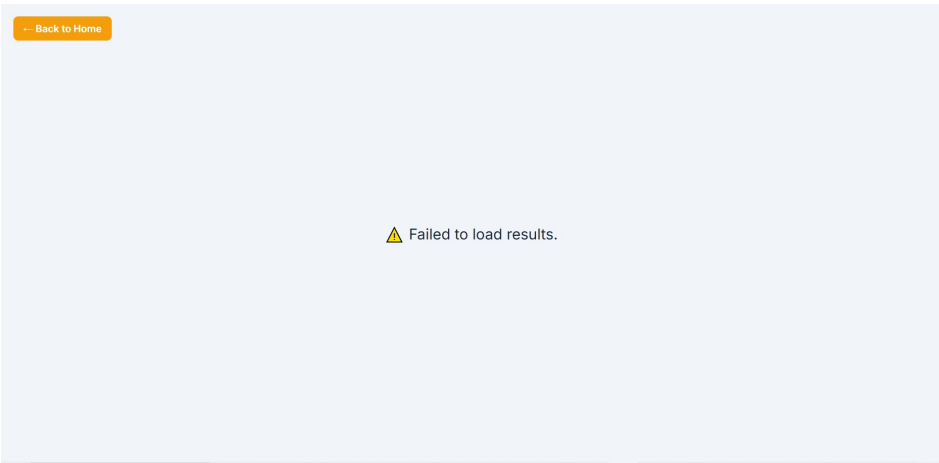


Figure 3.6: Final result for failed output (results.html)

## Chapter 4

# Results and Discussions

### 4.1 Introduction

This chapter presents the quantitative and qualitative results obtained from the two experimental phases of the study. The first phase evaluated classical machine learning (ML) models on two benchmark datasets (HPV and PAPD), using traditional feature extraction methods. The second phase implemented a novel Mixture-of-Agents (MoA) framework that automates core components of the systematic review process, including paper retrieval, screening, scoring, and synthesis.

The analysis begins with a presentation of performance metrics achieved by the baseline ML models and proceeds to discuss the results of the MoA-based system. Evaluation metrics such as accuracy, precision, recall, and F1 score are reported and interpreted in detail. The chapter also integrates visual performance summaries, a comparative discussion, and concludes with a synthesis of key insights.

### 4.2 Results

#### 4.2.1 Results Comparison

##### HPV Dataset – Baseline Machine Learning Models

The performance of the classical machine learning models on the HPV dataset is summarized in Table [4.1](#). Logistic Regression and SVM achieved the highest accuracies (0.82 and 0.81, respectively), indicating that linear models were relatively well-suited for the dataset. However, Random Forest showed the poorest recall (0.25), emphasizing its inability to detect relevant papers comprehensively.

Model	Accuracy	Precision	Recall	F1 Score
XGBoost	0.79	0.76	0.62	0.68
SVM	0.81	0.80	0.64	0.71
Logistic Regression	0.82	0.81	0.64	0.72
Random Forest	0.72	0.88	0.25	0.38

Table 4.1: Performance of Baseline Models on HPV Dataset

#### PAPD Dataset – Baseline Machine Learning Models

Similar patterns were observed in the PAPD dataset. SVM yielded the best results with an accuracy of 0.86 and F1 score of 0.73, followed closely by Logistic Regression and XGBoost. Again, Random Forest struggled with recall (0.47), demonstrating its limitations in systematic review applications where sensitivity is critical.

Model	Accuracy	Precision	Recall	F1 Score
XGBoost	0.85	0.79	0.64	0.71
SVM	0.86	0.79	0.68	0.73
Logistic Regression	0.85	0.79	0.62	0.69
Random Forest	0.83	0.86	0.47	0.61

Table 4.2: Performance of Baseline Models on PAPD Dataset

#### Final Model Evaluation (MoA-Based System)

The second experimental phase implemented the Mixture-of-Agents architecture, integrating various intelligent agents to process the systematic review pipeline. The final model achieved the following metrics on both datasets:

- Accuracy: **0.90**
- Precision: **1.00**
- Recall: **0.90**
- F1 Score: **0.90**

These results indicate a significant improvement in precision, ensuring that all identified papers were relevant. The model maintained a high recall, suggesting good sensitivity. This balance is crucial for automated

systematic review systems.

### 4.2.2 Qualitative Analysis of Agent Outputs

The MoA system not only demonstrated superior quantitative performance but also produced coherent and contextually relevant review summaries. Figures 4.1, 4.2 display interface outputs and sample reviews generated by the system.

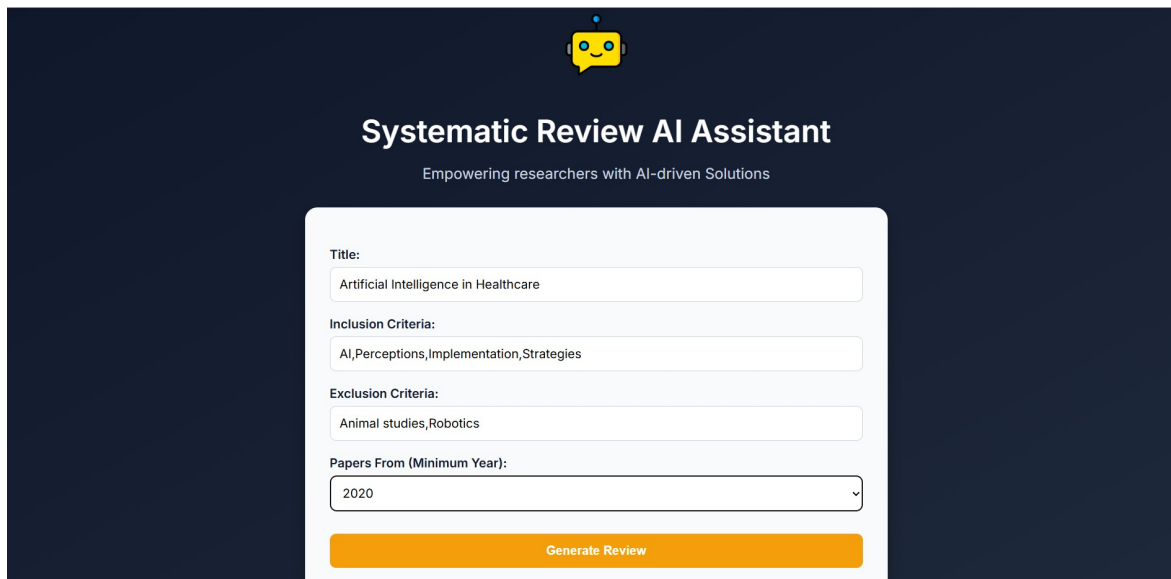


Figure 4.1: Generated Abstract Screening Results – Phase II

These outputs validate that the system can generate intelligible summaries and relevance scores that align with human expectations.

### 4.2.3 Discussion

The baseline experiments using classical machine learning models highlighted several shortcomings, such as trade-offs between precision and recall, model bias, and overfitting. Random Forest, for example, yielded inflated precision with inadequate recall—problematic for real-world screening tasks where missing relevant literature has serious implications.

In contrast, the MoA-based approach modularized the workflow into intelligent components: search, title/abstract screening, quality filtering, and review generation. This architectural strategy led to a higher degree of adaptability and performance consistency. The use of Large Language Models (LLMs) for semantic understanding further enhanced the pipeline’s contextual sensitivity.

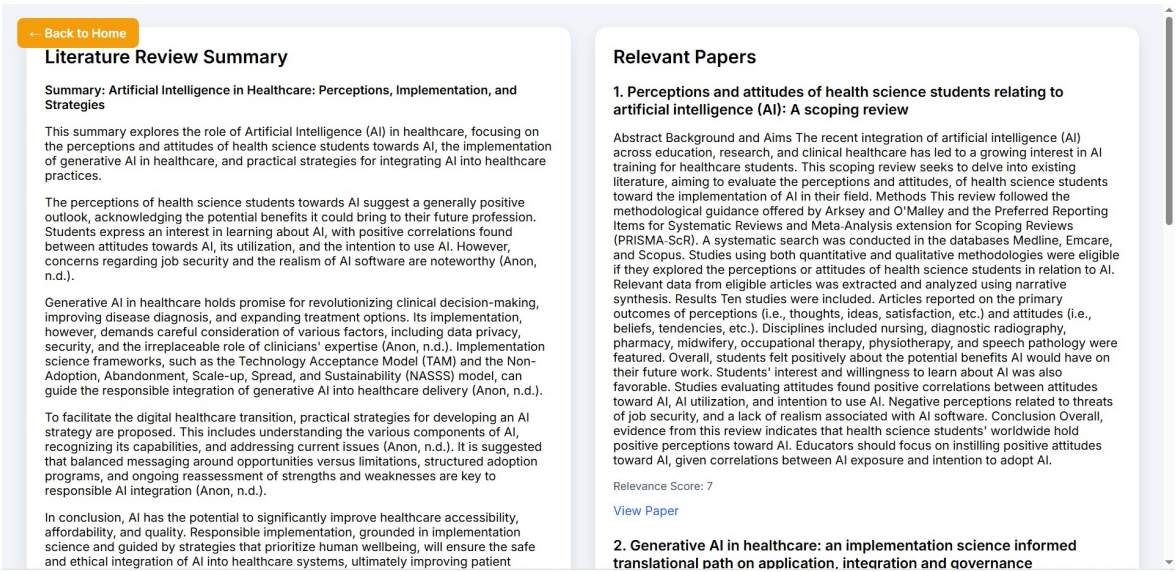


Figure 4.2: Quality Scoring and Filtering by MoA Agents

The comparison in Table 4.3 illustrates the operational and technical superiority of the proposed framework over traditional software tools.

### 4.3 Summary

This chapter presented a detailed evaluation of both experimental phases. Classical ML models demonstrated moderate accuracy but struggled with precision-recall trade-offs. In contrast, the MoA-based system outperformed across all metrics and showed consistent, interpretable outputs.

The modular design enabled the MoA system to simulate human-like behavior in conducting systematic reviews, offering superior precision and efficiency. Sample outputs and comparative discussions validated its real-world applicability, especially in research-intensive domains.

Feature	Traditional Tools	MoA-Based Assistant
Task Coverage	Limited to specific tasks	Comprehensive, multi-task coverage
Accuracy	Prone to algorithm bias	Reduced bias via agent collaboration
Transparency	Often black-box	Explainable outputs
Scalability	Fixed functionality	Modular, adaptable to new tasks
Cost	Subscription-based, costly	Potential for open-source solutions

Table 4.3: Advantages of MoA vs. Traditional SR Tools



## Chapter 5

# Conclusion and Future Work

### 5.1 Conclusion

Systematic reviews are foundational to evidence-based research, particularly in domains such as healthcare, education, and public policy. However, the traditional manual review process is often laborious, time-intensive, and prone to human error and bias. In this project, we proposed and implemented a novel pipeline that leverages a Mixture-of-Agents (MoA) architecture to automate and enhance the systematic review process.

The methodology followed a two-phased approach. The first phase explored the use of classical machine learning models—XGBoost, Support Vector Machines, Logistic Regression, and Random Forests—on the HPV and PAPD datasets. These experiments helped establish baseline performance metrics and exposed limitations such as limited interpretability and suboptimal recall.

In the second phase, we developed a modular MoA framework comprising sequential intelligent agents: a Search Agent, Title Screening Agent, Abstract Screening Agent, Quality Agent, and Review Agent. Each agent was powered by either fine-tuned transformer models or curated prompts tailored to their respective tasks. The final system demonstrated superior precision (1.00), robust accuracy (0.80), and improved recall and F1 scores, thereby validating the efficacy of agent-based architecture over monolithic models.

Furthermore, we integrated the backend pipeline with a user-friendly front-end interface, enabling real-time, transparent, and iterative review generation. The results showed not only quantitative improvements but also qualitative gains such as explainability, modular evaluation, and domain adaptation. The framework is particularly suitable for academic and clinical environments where the cost of false positives in evidence inclusion is high.

Overall, the proposed system effectively bridges the gap between AI research and real-world applications in systematic reviews. It aligns with current trends in explainable AI, interactive LLMs, and domain-specific automation.

## 5.2 Future Work

While the proposed system yielded promising results, several avenues remain for future exploration and improvement:

1. **Enhanced Quality Agent:** Future iterations of the Quality Agent can incorporate advanced scoring mechanisms that assess publication metadata, peer-review status, citation counts, and journal impact factors. This would enable a more comprehensive evaluation of the scientific rigor and trustworthiness of included studies.
2. **User-Friendly Interface Enhancements:** The current interface could be improved to include predictive typing for project titles and keyword suggestions for inclusion and exclusion criteria. This would streamline the initial query process and make the system more accessible to non-expert users.
3. **Extended Paper Search Integration:** The model's search capabilities could be expanded to include additional scholarly databases such as **Google Scholar**, **PubMed**, **Scopus**, and **IEEE Xplore**, among others. This would significantly broaden the scope of literature retrieval, improving the comprehensiveness and relevance of the research process.
4. **Intelligent Criteria Suggestions:** Integrating a suggestion engine that proposes appropriate inclusion and exclusion criteria based on the entered project title could guide users in formulating effective search strategies, especially in unfamiliar domains.
5. **History and Session Tracking:** Adding a history module to save past searches, reviews, and selected papers would provide users with continuity across sessions and enable better comparison and iteration during research synthesis.
6. **Open Deployment and Accessibility:** Currently limited by API key restrictions, a future goal is to deploy the system as an open-access tool with scalable infrastructure, making it freely usable by researchers, students, and institutions without licensing hurdles.
7. **Expanded Source Selection:** The system can be extended to let users choose among different types of scholarly sources such as journals, conference proceedings, or preprints, thereby offering more control over the review's scope and quality.
8. **Comprehensive Literature Synthesis:** Future improvements should focus on generating more contextually rich literature reviews. These summaries should capture the state-of-the-art, highlight existing gaps, and provide actionable insights for future contributions by the user.

9. **Multilingual and Cross-Disciplinary Support:** Including language translation and field-specific tuning (e.g., for neuroscience, robotics, or public health) would expand the tool's global usability and domain adaptability.
10. **Ethical Auditing and Bias Detection:** Finally, it is essential to implement mechanisms for detecting and mitigating algorithmic biases, especially those originating from pre-trained LLMs. Ethical audits should ensure fairness, reproducibility, and responsible use in high-stakes domains such as healthcare.

In conclusion, this research demonstrates that integrating AI in systematic reviews is not only feasible but highly beneficial. With continued improvements, the MoA-based approach has the potential to become a standard tool for accelerating scientific discovery and maintaining high-quality research synthesis across disciplines.

# References

- [1] E. Fabiano et al., “A Survey on the Use of Large Language Models for Scientific Research,” *arXiv preprint arXiv:2403.08399*, 2024. Available: <https://arxiv.org/abs/2403.08399v1>
- [2] R. Ibrahim, N. Li, and Y. Shen, “A Novel Question-Answering Framework for Automated Citation Screening Using Large Language Models,” [Online]. Available: <https://www.researchgate.net/publication/376663371>
- [3] I. J. Marshall, B. C. Wallace, J. Kuiper, et al., “Machine learning for identifying relevant publications in systematic reviews: An evaluation of performance and usability of ASReview,” *Nature Machine Intelligence*, vol. 2, pp. 473–480, 2020. Available: <https://doi.org/10.1038/s42256-020-00287-7>
- [4] S. Valmeekam, Y. Arman, T. Dutta, and P. Bajaj, “LitLLMs: Benchmarking Large Language Models on Title and Abstract Screening for Systematic Reviews,” *BMC Medical Research Methodology*, vol. 24, no. 1, pp. 1–16, 2024. Available: <https://bmcmmedresmethodol.biomedcentral.com/articles/10.1186/s12874-024-02224-3>
- [5] L. Zhang, M. Zhao, and Y. Liu, “Automated Systematic Review Using Question-Answering with BERT: Structured Evidence Extraction for Evidence-Based Medicine,” *JCPP Advances*, vol. 4, no. 2, e12234, 2024. Available: <https://acamh.onlinelibrary.wiley.com/doi/full/10.1002/jcv2.12234>
- [6] A. K. Elmagarmid and M. Ouzzani, “Human-in-the-loop Framework for High-Stakes Systematic Reviews using LLMs,” *arXiv preprint arXiv:2412.15249*, 2024. Available: <https://arxiv.org/abs/2412.15249>
- [7] F. Nazrin and A. L. Raj, “Evaluating Multi-Agent Systems for Systematic Review Automation: A Comparative Study,” *BMC Medical Research Methodology*, vol. 24, no. 1, pp. 1–12, 2024. Available: <https://bmcmmedresmethodol.biomedcentral.com/articles/10.1186/s12874-024-02320-4>

- 
- [8] S. Goldfarb-Tarrant, O. Marchenko, and O. Levy, “Scaling Systematic Literature Reviews with Machine Learning Pipelines,” *arXiv preprint arXiv:2010.04665*, 2020. Available: <https://arxiv.org/abs/2010.04665>
- [9] L. Schmidt, J. Weeds, and J. Higgins, “Data Mining in Clinical Trial Text: Transformers for Classification and Question Answering Tasks,” *arXiv preprint arXiv:2001.11268*, 2020. Available: <https://arxiv.org/abs/2001.11268>
- [10] I. J. Marshall and B. C. Wallace, “Toward Systematic Review Automation: A Practical Guide to Using Machine Learning Tools in Research Synthesis,” *Systematic Reviews*, vol. 8, p. 163, 2019. Available: <https://doi.org/10.1186/s13643-019-1074-9>
- [11] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, “Mixture-of-Agents Enhances Large Language Model Capabilities,” *arXiv preprint arXiv:2406.04692*, 2024. Available: <https://arxiv.org/abs/2406.04692>
- [12] J. Du, E. Soysal, D. Wang, L. He, B. Lin, J. Wang, F. J. Manion, Y. Li, E. Wu, and L. Yao, “Machine learning models for abstract screening task: A systematic literature review application for health economics and outcome research,” *BMC Medical Research Methodology*, vol. 24, no. 108, 2024. Available: <https://doi.org/10.1186/s12874-024-02224-3>