Agentic AI for Automated Systematic Literature Reviews

Seminar Software Engineering

Students Lino Meister
Wenhui Yu

Supervisor Prakash Aryan

SOFTWARE ENGINEERING GROUP
INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF BERN

Introduction

Systematic Literature Reviews

SYSTEMATIC LITERATURE REVIEW & PRISMA

- **SLR:** a structured and reproducible process where at least two researchers independently evaluate all relevant literature to answer a defined research question.

- **PRISMA 2020 Statement:** standardized guidelines for conducting and reporting SLR
  - Statement Paper: 27-item checklist and flow diagram.
  - Development Paper
  - Explanation & Elaboration Paper

Challenge & Idea

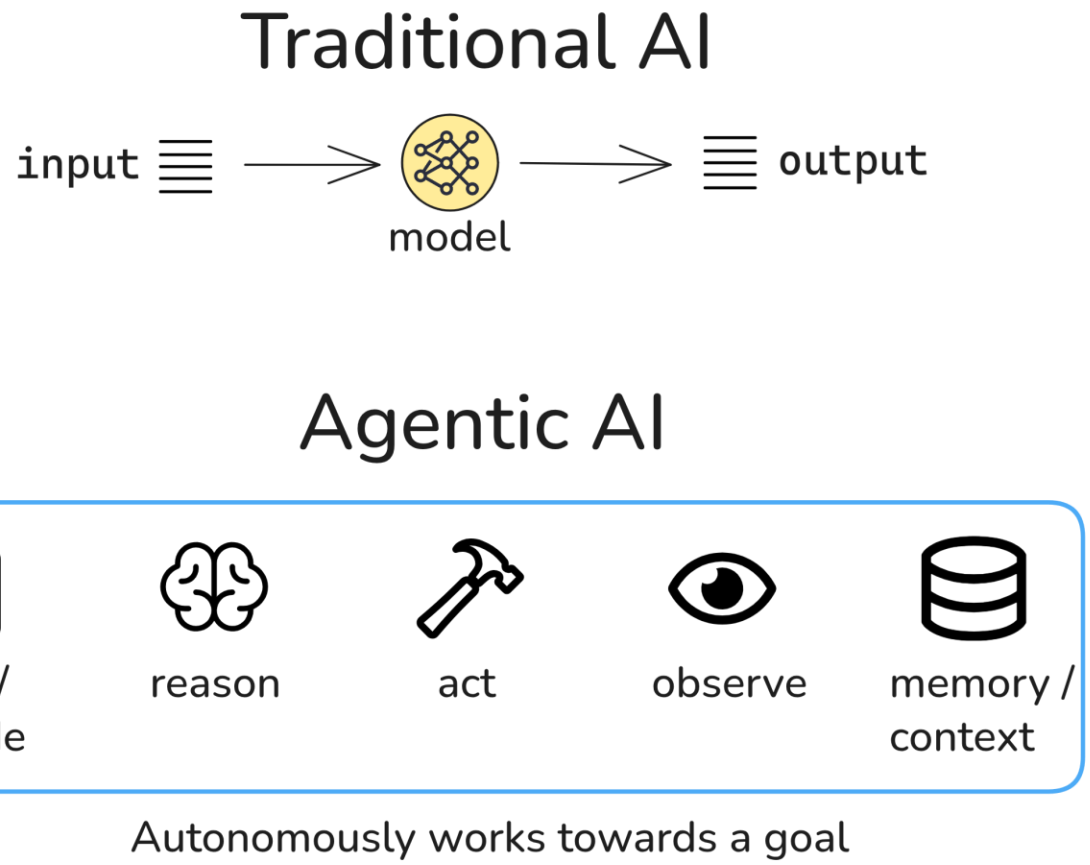
- **The Challenge of manual SLRs:**
  - **Volume:** thousands of candidate papers
  - **Time:** screening and citation take months

- **The idea:**
  - Use agentic AI to accelerate the workflow while the researcher always makes the final decisions through Human-in-the-Loop checkpoints.

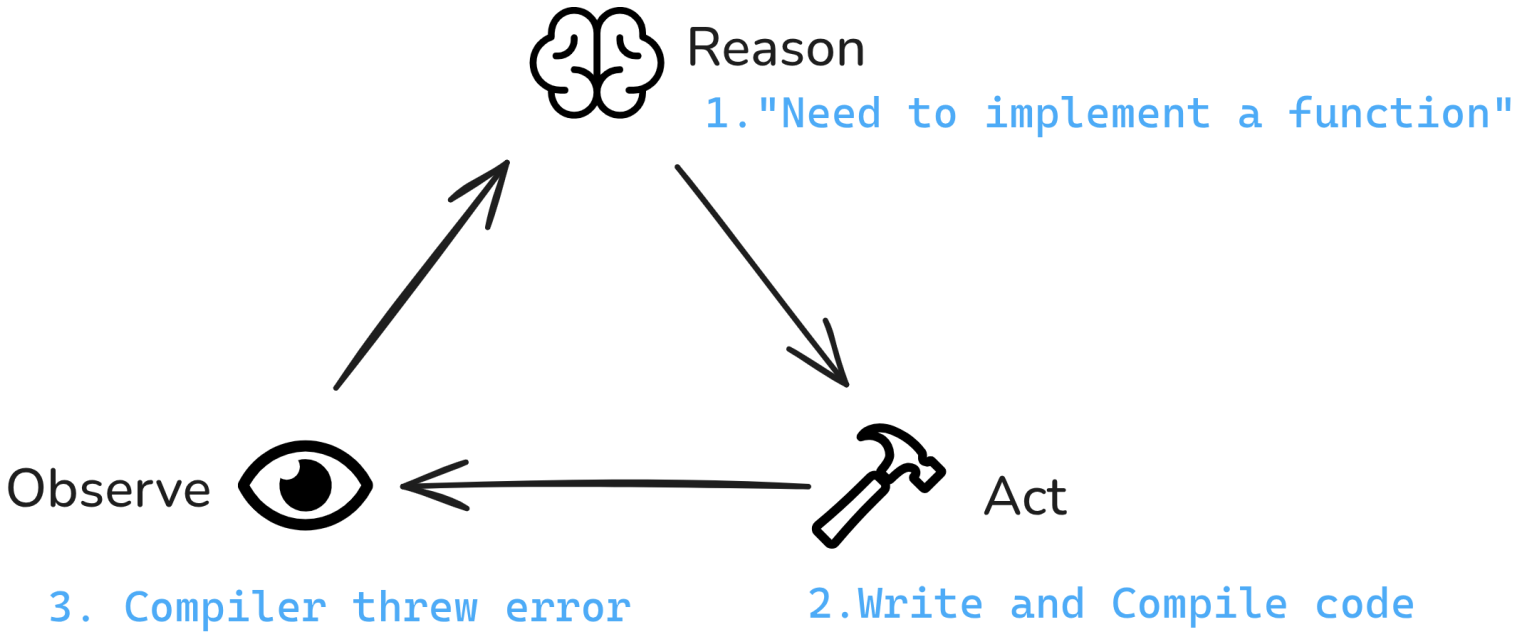
Introduction

Agentic AI

Agentic AI


Traditional AI
input
model
output
Agentic AI
plan / decide
reason
act
observe
memory / context
Autonomously works towards a goal

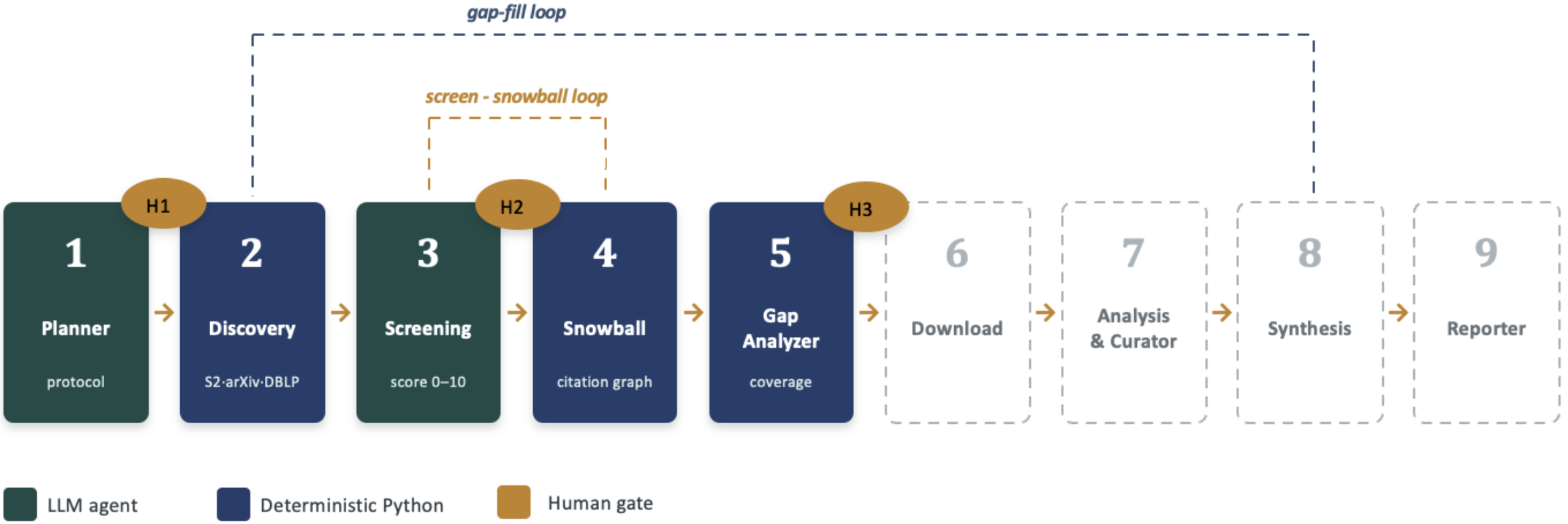
ReAct Pattern



Agentic AI for Automated Systematic Literature Reviews

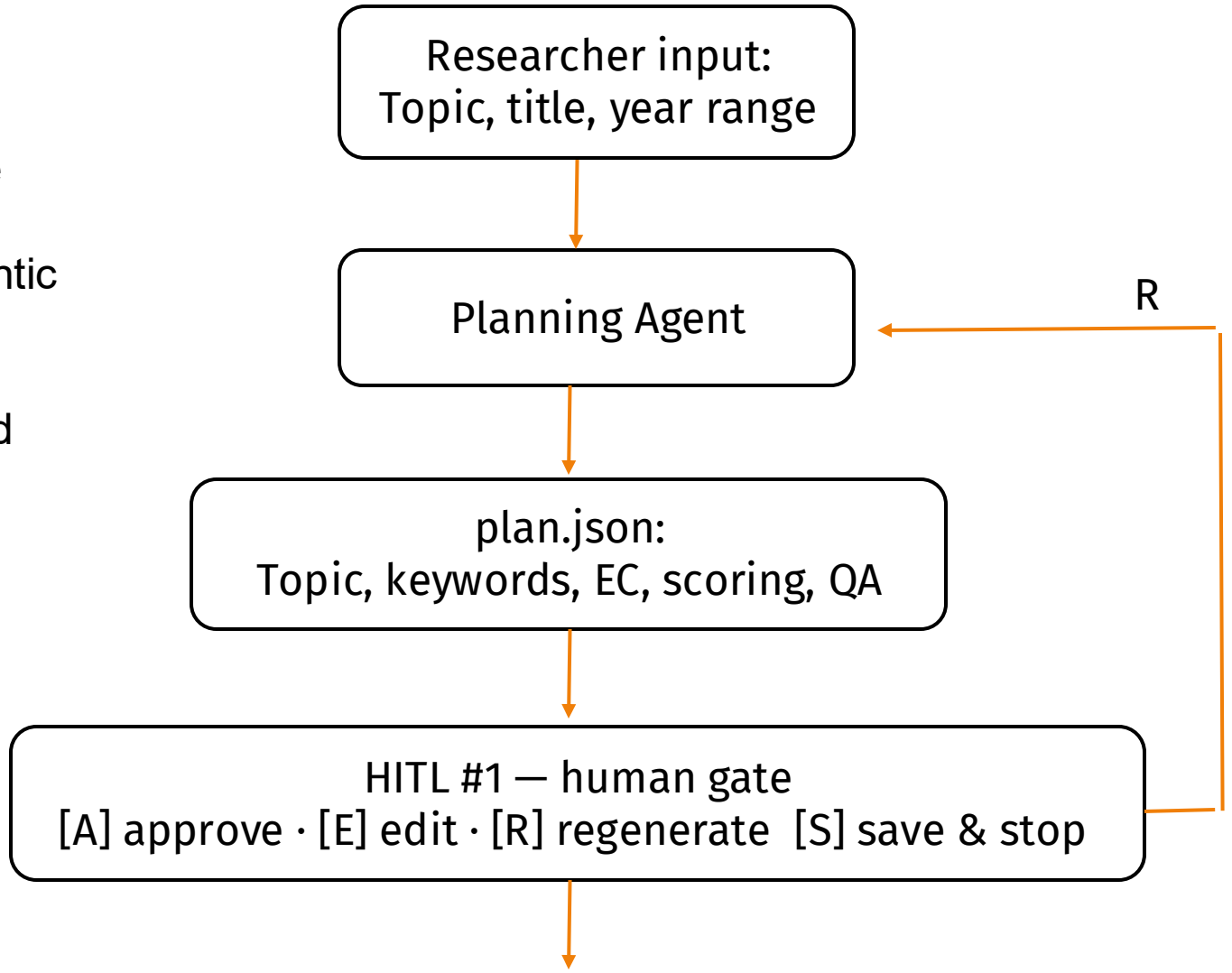
(Part 1)

Pipeline Part 1



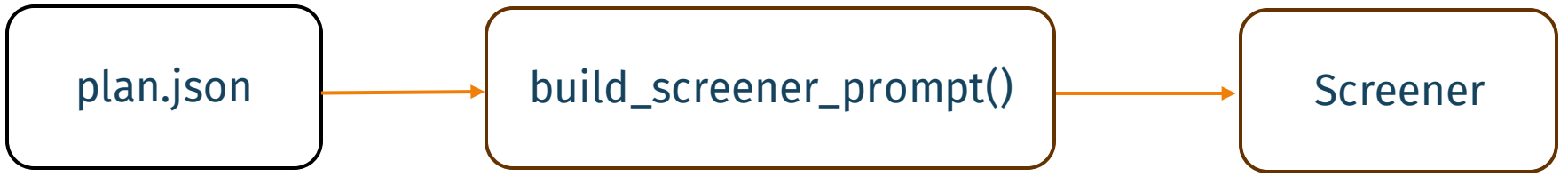
Planner

- **One Central Plan:** Generate plan.json, SSoT
- **Output Control:** Uses Pydantic and a retry loop to fix wrong LLM formats.
- **HITL # 1:** presents generated plan to Researcher, approve the rules before execution.

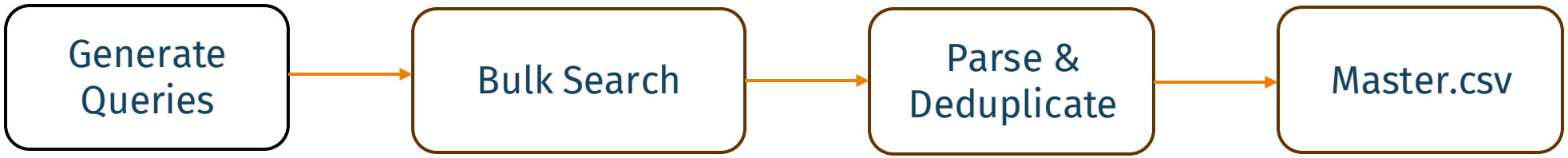


Prompt Builder

- **Separate prompt construction from agent implementation:** Agent code contains zero project-specific logic. Eg. exclusion criteria are loaded into the prompt from plan.json at screening runtime.
- **Reusability:** Change the entire research domain by just updating the plan file.



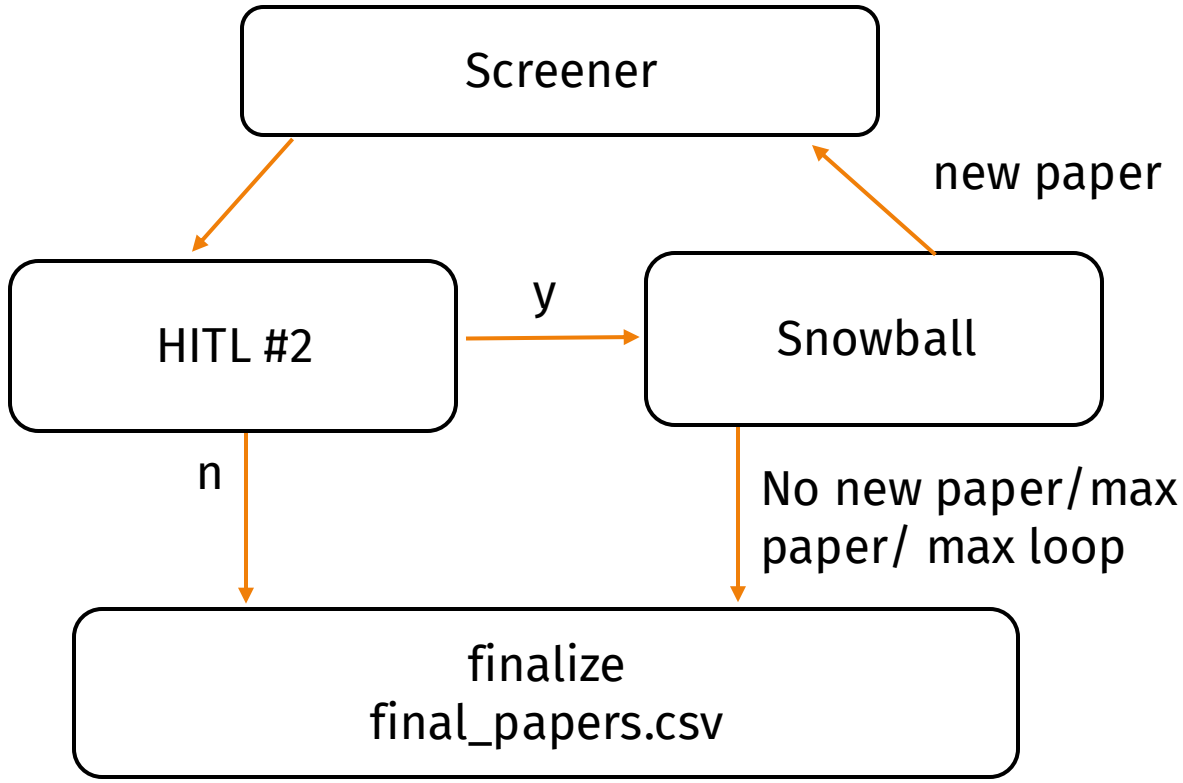
Collector


Generate Queries
Bulk Search
Parse & Deduplicate
Master.csv

Screener

Evaluates paper titles and abstracts according to the scoring rubric generated during the planning stage

- **Batch of 10**: depends on the model's context window.
- **Active Learning:** from second iteration includes prior High/Low samples in the next prompt



HITL #2 & Snowball

HITL #2:

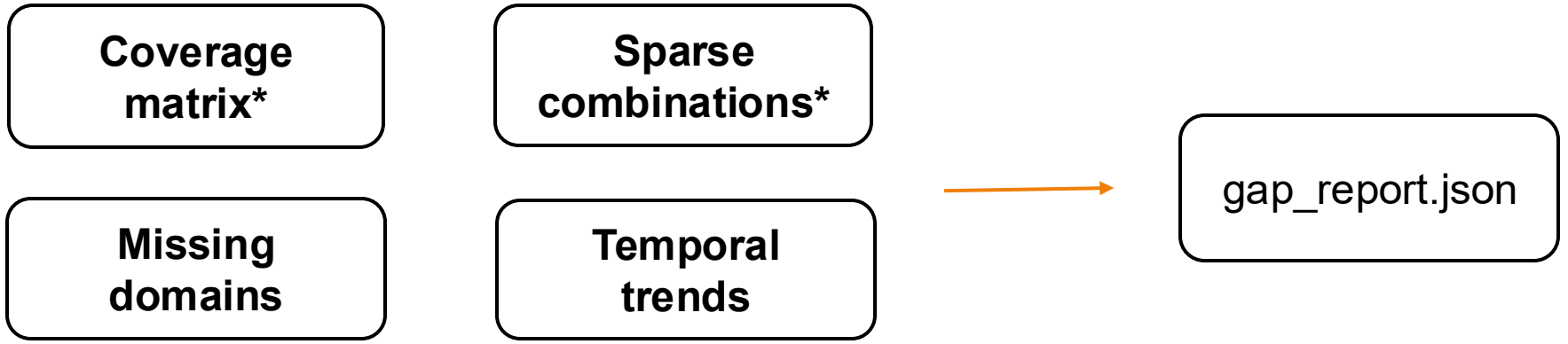
- presents a summary of the results, including number of accepted papers, rejected papers, and borderline cases
- human confirm "Continue snowball?" y / n

Snowball:

- takes the top-8 papers scoring ≥ 7.5 as seeds, pulls their backward references and forward citations.
- Stop condition: when no new paper or predefined maximum limit of papers
- New papers are merged into the unscored list for the next round of screening

Gap analyzer & HITL #3

Four deterministic analyses, no LLM — every result is explainable and reproducible.



HITL #3:

- present the gap report and download list
- approves the download list

**Concept matrix — Webster & Watson (2002)*

Agentic AI for Automated Systematic Literature Reviews

(Part 2)

Traditional SLR Stages (Part 2)

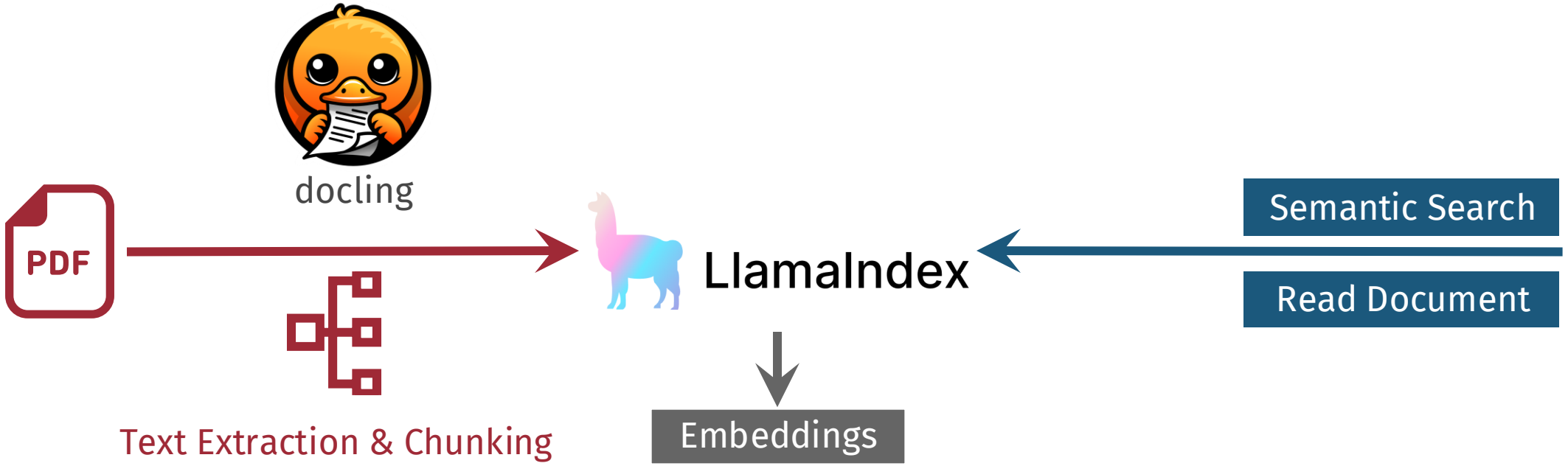
Human reviewers

- Read all the papers and **extract** key information
- **Exclude** irrelevant or poor-quality papers
- **Synthesize** their findings into a **report**

Agentic AI SLR Stages (Part 2)

- Preparation: Make papers processable
- Review: Extract key information and exclude irrelevant papers
- Synthesis: Aggregate the per-paper information
- Report: Write the report

Knowledge Base



Review

Process the entire content of the paper, extract **properties**, evaluate the **relevancy** and **quality** of the papers to **filter out** irrelevant ones.

We expect to be given:

- A few hundred papers (pdf + metadata)
- A context specifying
  - What properties to extract
  - Research questions
  - Quality / relevancy criteria

Review

Extraction

- Algorithms (A, B, C)
- Problems (X, Y, Z)

Inclusion / Exclusion

RQ1: What is the best approach for solving X?
…
QA1: Contains empirical evaluation
…
IC1: …
EC1: …

Extraction

- Algorithms: A,B
- Problems: X

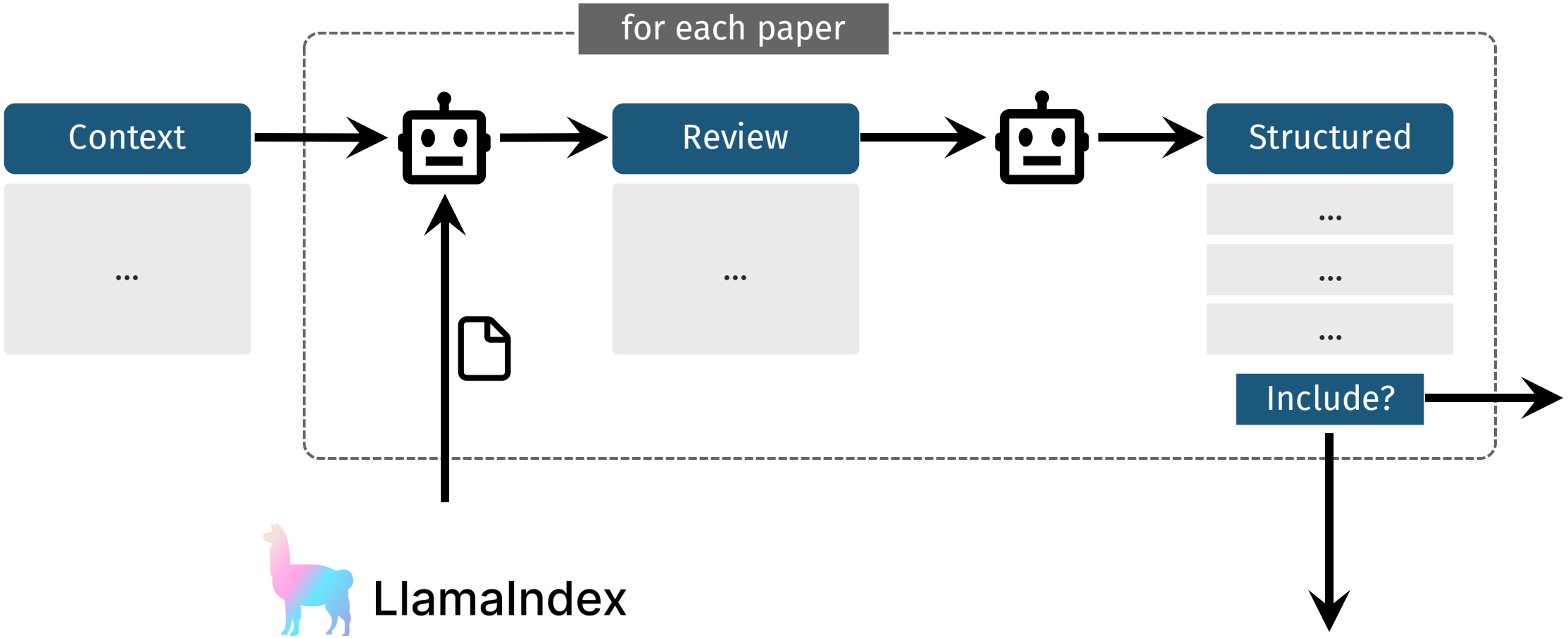
Inclusion / Exclusion

RQs: RQ1, contribution
QAs: QA1
ICs / ECs: IC1
Quality Score: 7

Additional

Findings: Algorithm *A*, faster than *B*
Limitations: *A* only works under assumptions
Major focus: Comparison of *A* and *B*

Review


for each paper
Context
...
Review
...
Structured
...
...
...
Include?
LlamaIndex

Synthesis

Aggregate the per-paper information and build the core knowledge foundation for the report

- Cluster papers into different **themes**
- Formulate answers to the **research questions**
- Construct **Artifacts**
  - Tables (e.g., method × application)
- Find **gaps** in the research
  - E.g., lack of recent research for a theme

Synthesis - Details

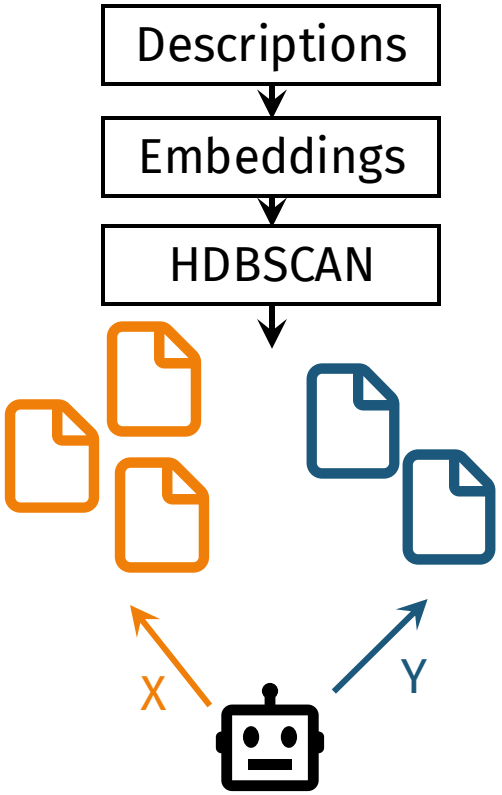
Research Questions

Can A solve X?

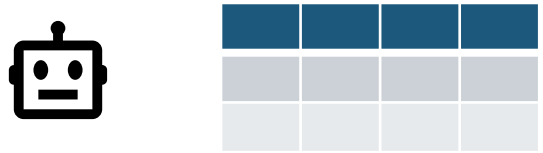
<Extractions from all papers addressing this research question>

Yes, because…

Clustering & Themes

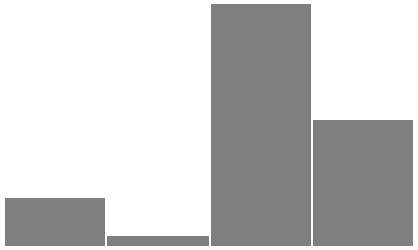


Artifacts

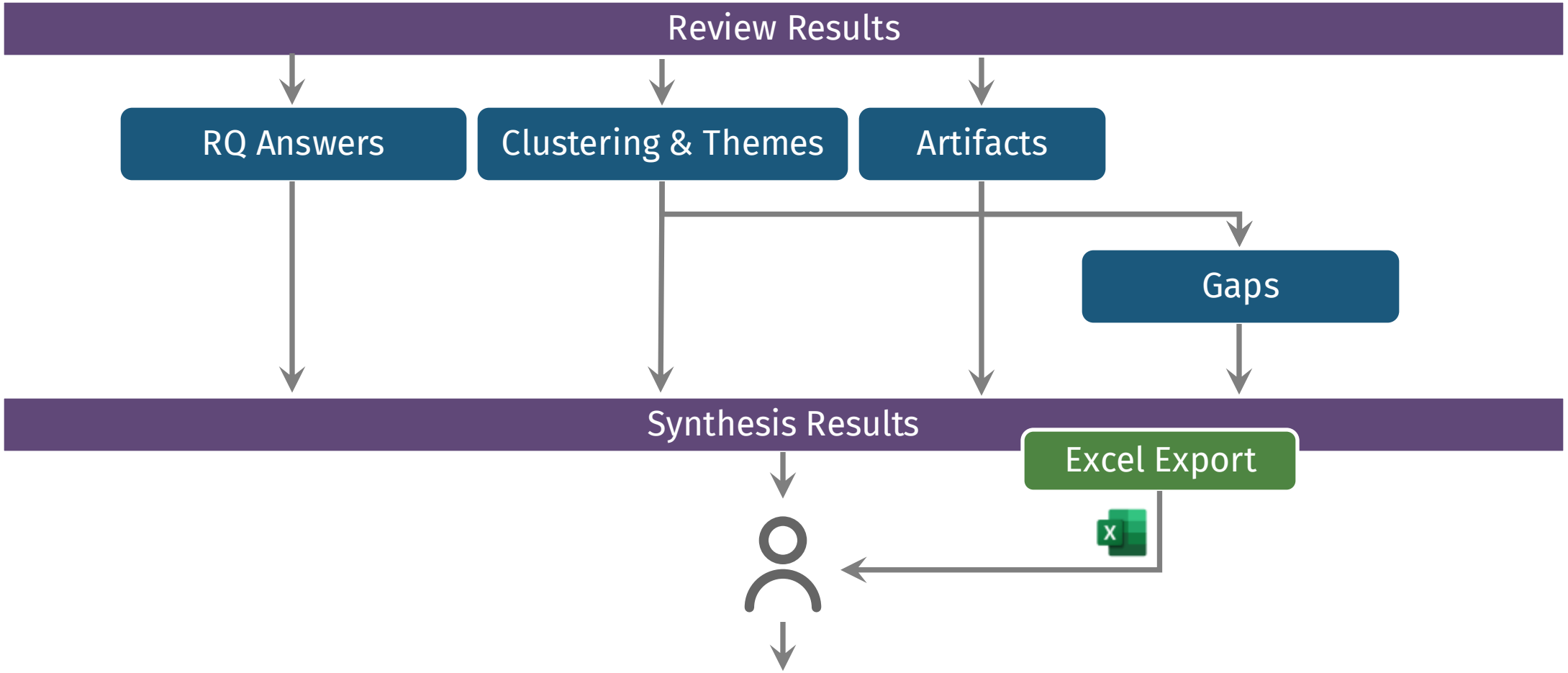


Algorithm × Problem Table

Gaps



Synthesis


Review Results
RQ Answers
Clustering & Themes
Artifacts
Gaps
Synthesis Results
Excel Export

Report

Produce a draft **report** of the review based on the synthesis results and export additional artifacts.

- Given a template with the report structure and instructions
- Produce a markdown draft report

Report

Template

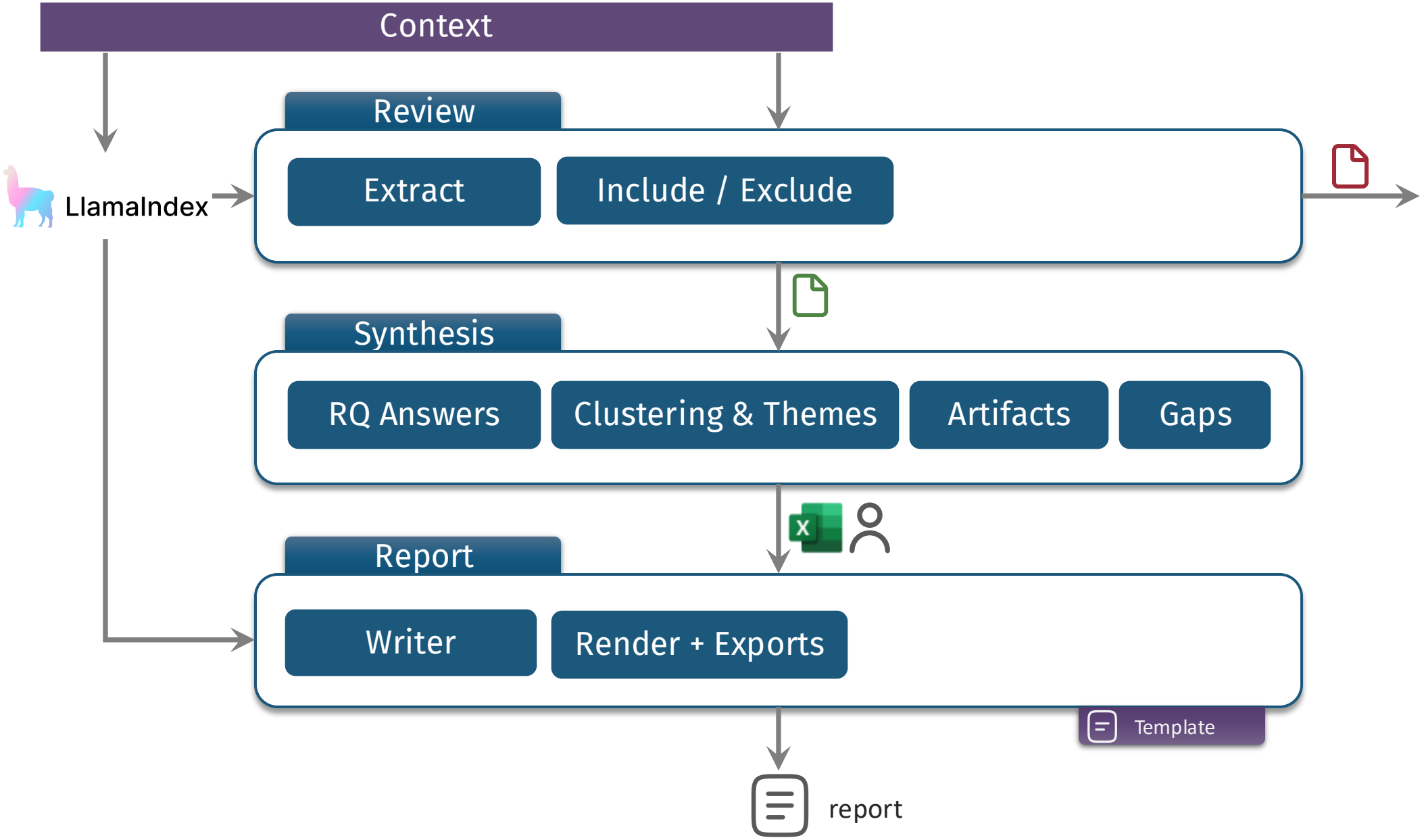
Synthesis Results

Writer Agent (+ Vector Search Tool)

Render Report: Insert artifacts, create markdown file

Exports: PRISMA diagram, bibliography

report.md


Context
Review
LlamaIndex
Extract
Include / Exclude
Synthesis
RQ Answers
Clustering & Themes
Artifacts
Gaps
Report
Writer
Render + Exports
Template
report

Results

Some example outputs

Models

For the test runs we used models on the universities gpustack

- LLM: qwen3-coder-30b-a3b-instruct
- VLM: qwen3-vl-30b-a3b-instruct
- Embeddings: qwen3-embedding-0.6b

Discussion

Limitations and Improvements

Future Improvement part 1

- **Sharper exclusion criteria**: define them more precisely to reduce borderline cases.
- **Abstract-only screening**: a full-text screening for uncertain papers.
- **Broader snowballing**: currently Semantic Scholar only — extend to other citation sources for better coverage.
- **Hybrid Download Prioritization Rule**
- The gap-fill loop to be implemented

Limitations - Knowledge Base

Docling text extraction

- ⚠️ Currently speed & stability prioritized over quality
- 🔧 OCR capabilities (e.g., for scanned papers)
- 🔧 Extracting formulas in LaTex format

Limitations - Review

Second Agent

- ⚠️ Currently just produces structured outputs
- 🔧 Could also act as verifier (e.g., detect missing or invalid properties)
- 🔧 Hand back to first agent if problem detected (or flag the paper)

Limitations - Review

Free-text extraction attributes (e.g., “extract method”)

- ⚠️ Can take many distinct values, not useful for artifacts
- 🔧 Either add post-processing step to cluster values
- 🔧 Or make agent select from predefined values

React to abnormal results

- 🔧 Report frequencies: Attributes, inclusions, RQs, QAs, scores
- 🔧 Add HITL interaction to react and re-run

Limitations - Synthesis

Themes

- ⚠️ Only have one pass of clustering + naming
- ⚠️ Clusters are looked at independently
- 🔧 Give agent full context, give agent control over clustering params

Gap Analysis

- ⚠️ Not automated with agent
- ⚠️ No gap fill loop

HITL

- ⚠️ No editing or flow control

Limitations - Report

Produces a rough markdown draft

- ⚠️ Only table artifacts, no plots generated

Context management

- ⚠️ Gets a large amount of information from previous stage
- 🔧 Orchestration agent
  - Determines subtask (e.g., writing section) and specific context
  - Performed by sub agents

Limitations – System Wide

⚠️ **Human Interactions & flow control**

- Limited human interactions
- No flow control

⚠️ **Quality evaluation**

- Compare against a «traditional» SLR
  - Inclusion / exclusion decision
  - Answers to research questions

Repository on GitHub

https://github.com/prakash-aryan/paper_collector_slr