Generative AI (WS 2025/2026)
Project Report

AI-Driven Multi-Research Paper Summarization and Knowledge-Graph Construction

Group 67
Angerer Robert
Bhardwaj Aman
Bilounga Gehrmann Jenny Stephanika
Enkhtuul Enkhjin
Marsh Linnea

Abstract

In modern research, the exponential growth of scientific literature makes it increasingly difficult for students and researchers to efficiently extract, synthesize, and utilize information. Reading multiple papers to identify connections, methods, datasets, or contributions is time-consuming and error-prone. Existing tools such as reference managers or basic search engines help with organization but fail to provide automated synthesis and structured knowledge representation.

So, through this project we are developing an AI-driven system designed to automatically summarize research papers and construct an interconnected knowledge graph. By leveraging large language models (LLMs) and metadata APIs, it will enable users to extract key information, understand relationships between papers, and visualize knowledge connections.

Description

Objective: Compare the extracted Knowledge Graphs (KGs) from three sources: Scratch (baseline), Mistral, and LLaMA models. The analysis includes node and edge statistics, overlaps, and graph structural metrics.

The system allows users to extract, visualize, and evaluate three Knowledge Graphs (KGs) from the same set of PDFs using different methods:

1. **Scratch KG** – Baseline KG.

2. **Mistral KG** – Generated using the Mistral 7B model.
3. **LLaMA KG** – Generated using the LLaMA 3 model.

The purpose of this analysis is to:

- Compare **graph sizes** (nodes and edges)
- Measure **node and edge overlaps**
- Compute **graph structural metrics** such as density, average degree, and largest connected components
- Provide **visual comparisons** via bar charts
- Analyze **pairwise and three-way overlaps**

Knowledge Graph Sizes:

KG	Nodes	Edges
Scratch	958	55
Mistral	130	135
LLaMA	114	139

Three-way Node and Edge Comparison:

Metric	Value
Common Nodes (All 3)	22
Unique Scratch Nodes	932

Unique Mistral Nodes	31
Unique LLaMA Nodes	15

Observations:

- Only 22 nodes are shared among all three KGs, indicating limited consensus across models.
- Baseline has a very large number of unique nodes (932), likely capturing detailed but sparse information.
- Mistral and LLaMA contribute fewer unique nodes (31 and 15 respectively), suggesting overlap between these LLMs is higher.

Edges:

Metric	**Value**
Common Edges (All 3)	0
Unique Scratch Edges	55
Unique Mistral Edges	54
Unique LLaMA Edges	58

Observations:

- No edges are common across all three KGs, indicating that the relationships captured by each method are highly model-dependent.
- Scratch’s edges are sparse, while Mistral and LLaMA produce dense but divergent edges.

Direct user-journey:

- Setup: Users install Ollama, pull the Mistral 7B model, Llama 3 and start the local API server. The server runs locally at: http://localhost:11434. They then prepare a Python environment with required packages and the SpaCy English model.
- Prepare Data: PDFs to be analyzed are placed in the designated folder. Both pipelines automatically process these files.
- Run Pipelines:
    - Scratch KG extracts entities and relations using SpaCy NER and dependency parsing. Outputs web_client_implementation/triples.json.
    - LLM KG queries Mistral 7B and Llama3 with a guided prompt to extract entities and edges. Outputs web-client/triples.json.
- Visualize: Users start the web client by runnig on cmd: \web-client>python -m http.server 8000 and opening the browser: http://localhost:8000 and view nodes and edges interactively in Cytoscape.js for both Mistral adn Llama. Use \web_client_implementation>python -m http.server 8000 and in browser: http://localhost:8000/index.html for the baselin KG visualization.
- Evaluate: Metrics such as graph structure, node and edge overlap, F1 score, and edge accuracy are computed, and visual plots are generated for comparison.

This workflow provides a streamlined, end-to-end experience from raw PDFs to analyzed, visualized, and evaluated knowledge graphs.

Technical Approach

We implemented two complementary methods for constructing knowledge graphs (KGs) from scientific literature.

Scratch KG: Uses SpaCy for named entity recognition and dependency parsing to extract entities and explicit syntactic relationships. No relation types are hardcoded, making it a fully data-driven baseline.

LLM KG (Mistral 7B): Uses Ollama to query the LLM with a prompt specifying entities and relation types. Outputs nodes and edges in JSON format, capturing abstract or implicit relationships that the Scratch KG may miss.

LLaMA KG – Generated using the LLaMA 3 model

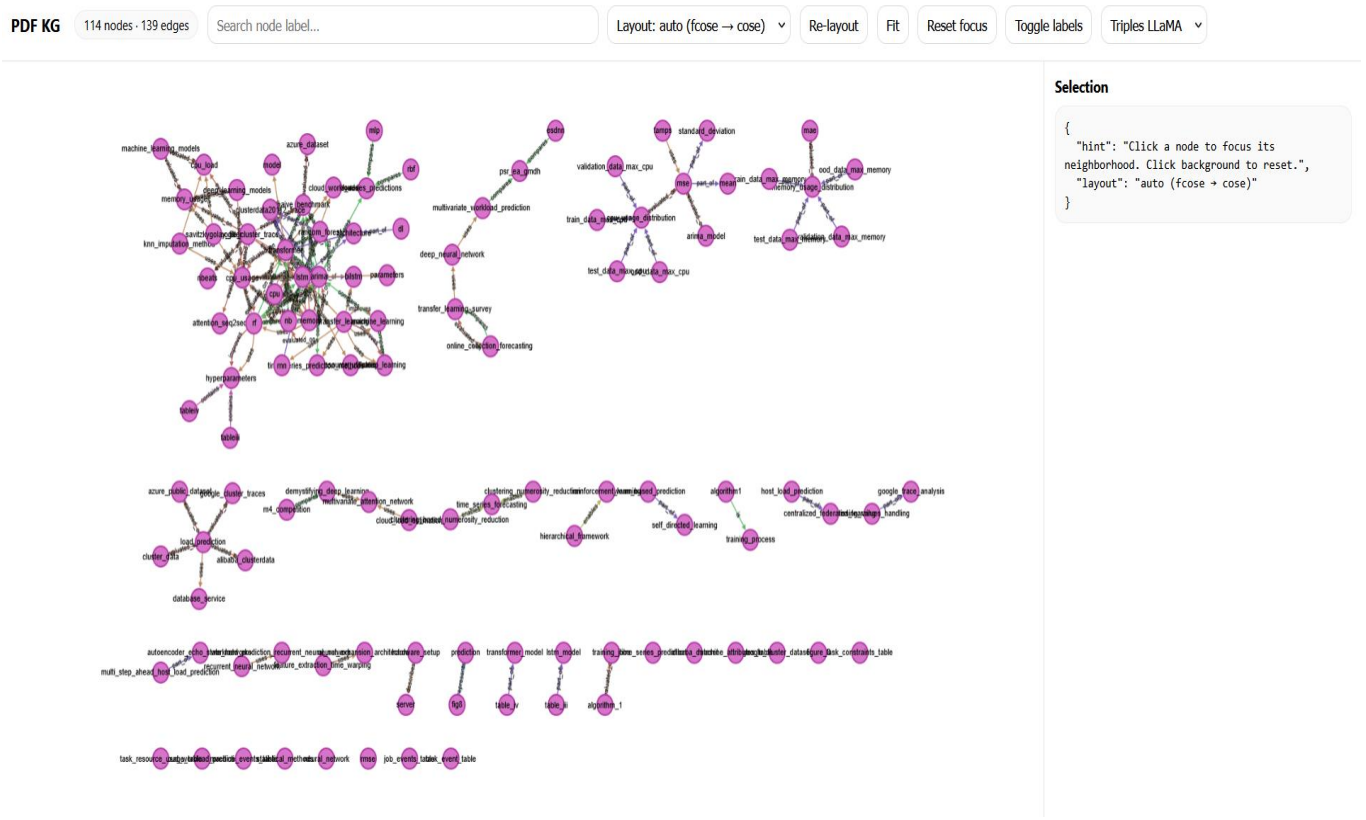
Evaluation Metrics: Graphs are compared by structure, node and edge overlap, and overall size. Edge F1 scores and node overlap ratios provide quantitative measures, while visualizations support qualitative analysis.

This approach allows comparison between precise, rule-based extraction and semantically rich LLM-based reasoning.

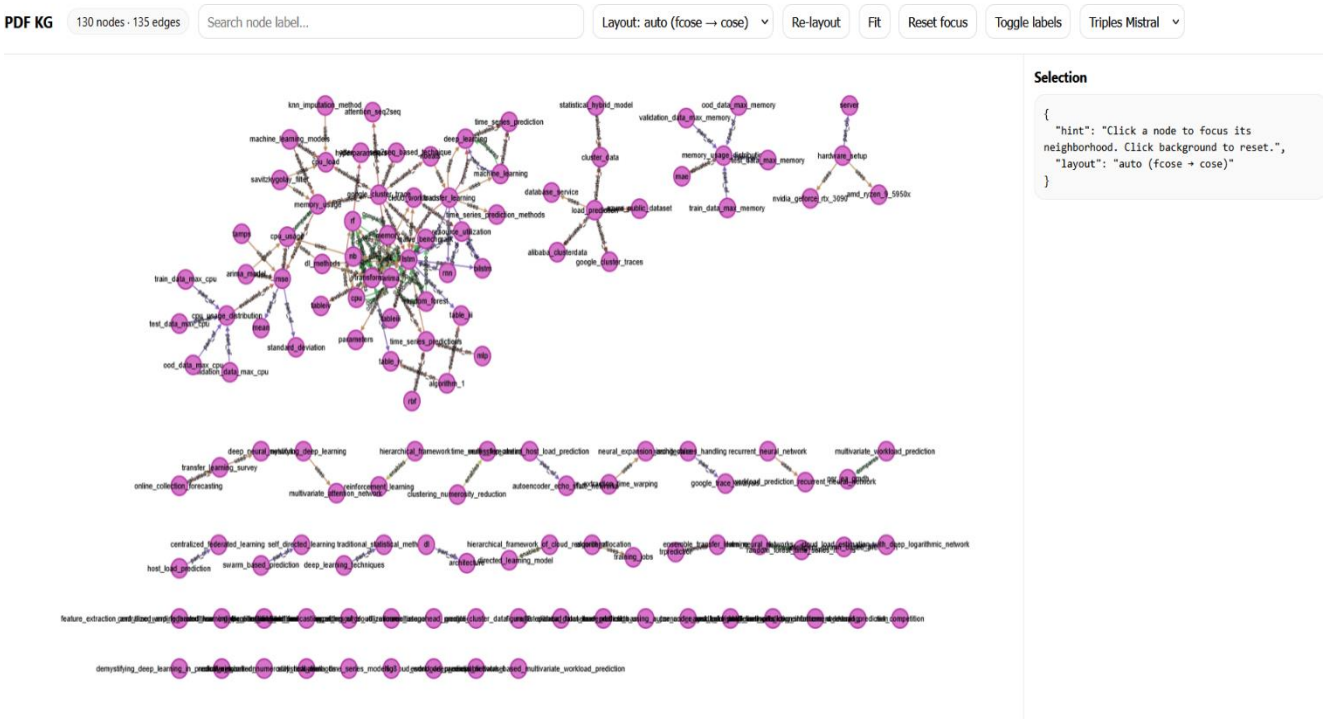
Challenges

One technical challenge we examined was that increasing the number of research papers led to significantly longer processing times for generating the knowledge graphs. To address this limitation, we ultimately restricted our analysis to two research papers and focused the evaluation of our system on these, comparing the results across two different models.

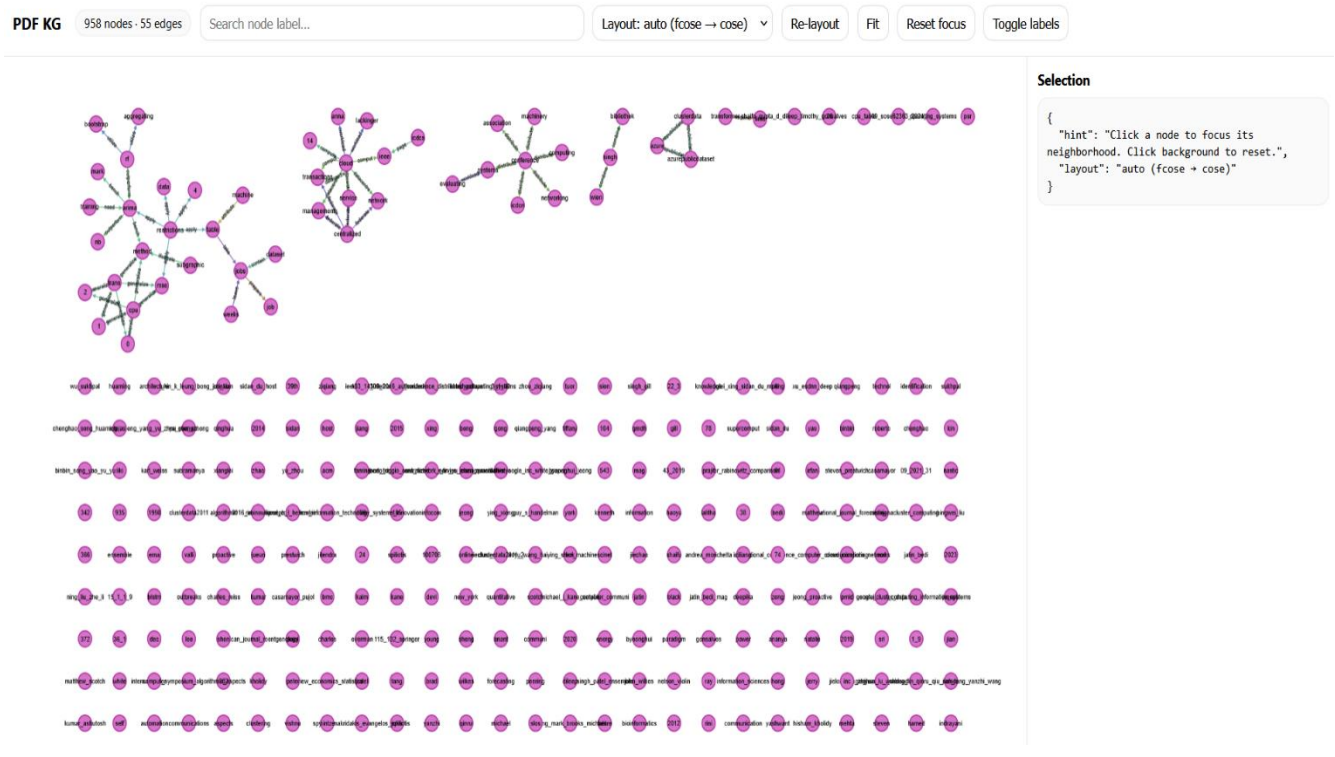
KG Visuals:



The above is the visualization of Knowledge graph generated using Llama.



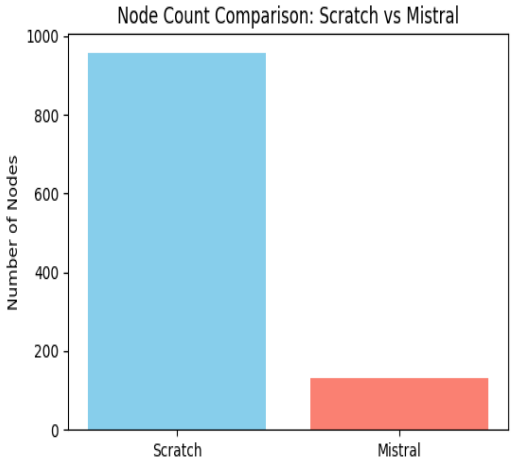
The above is the visualization of Knowledge graph generated using Mistral.

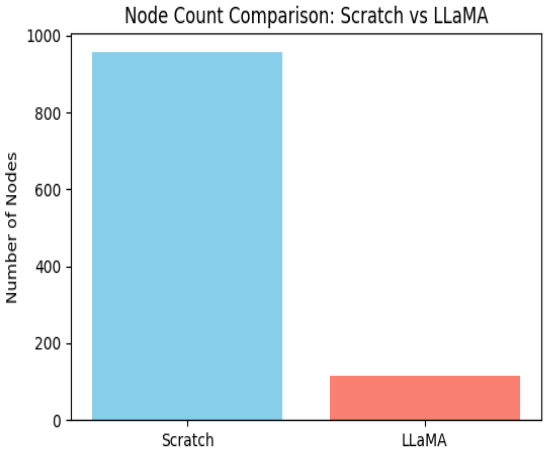


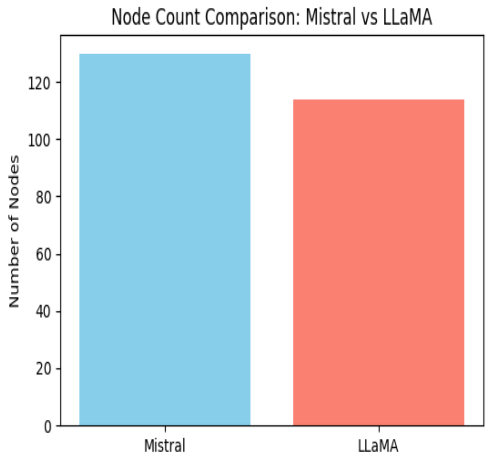
The above is the visualization of Knowledge graph generated using Baseline imlementation.

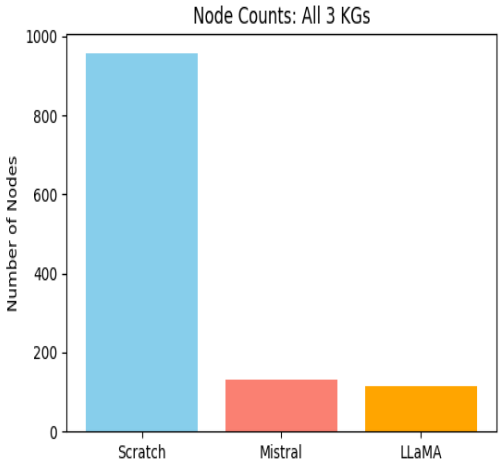
---- It can be seen clearly that the baseline was not able to make good relations between nodes and edges and performed poorly.

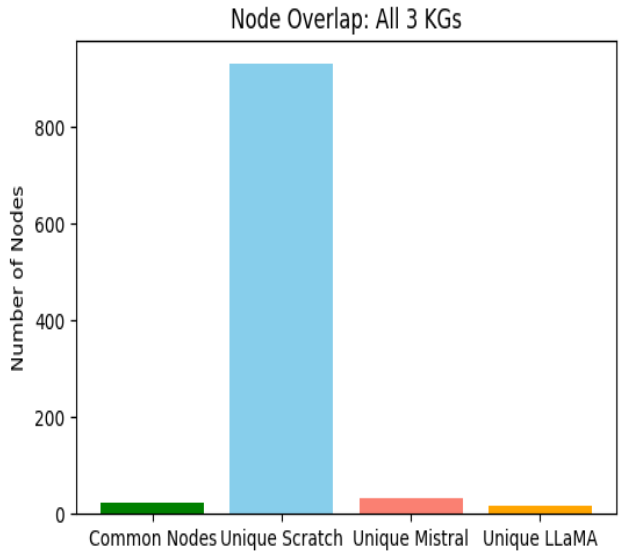
Evaluation and plots:

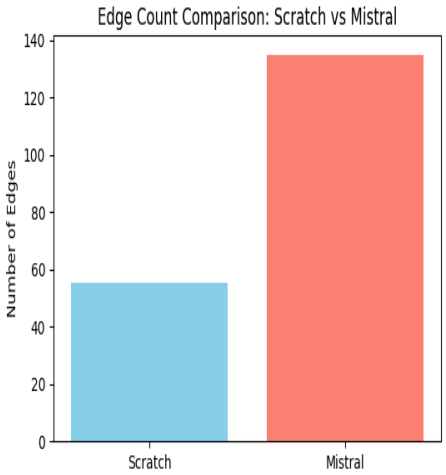


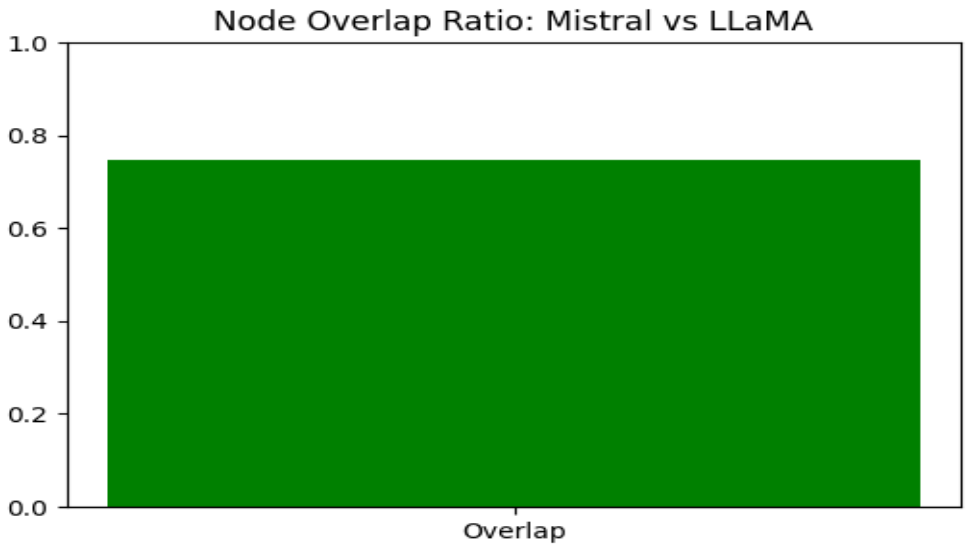








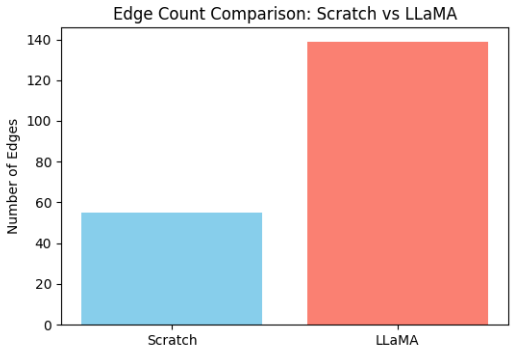


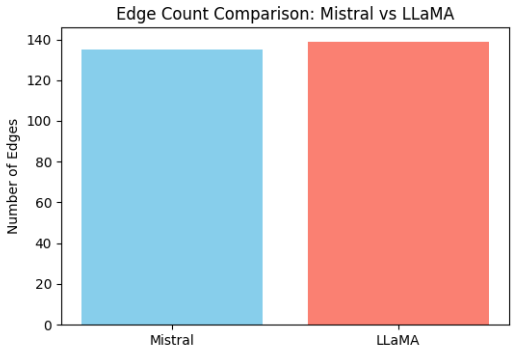


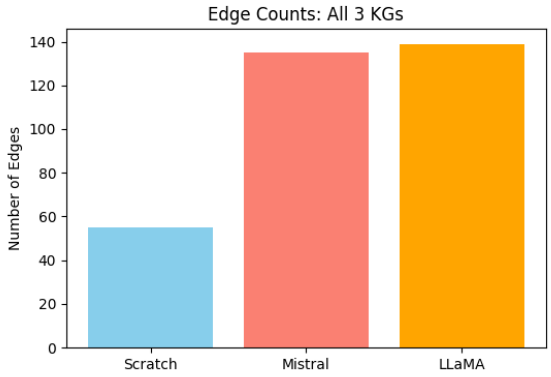
So from the above and below plots and results, we made the following observations as:

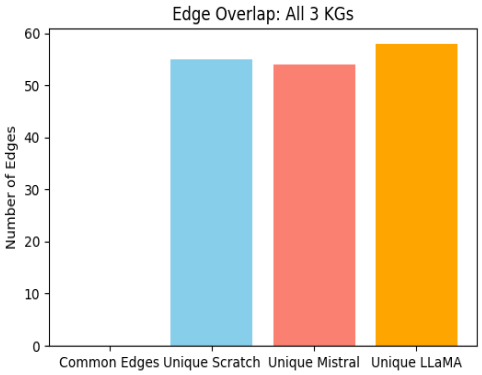
Observations (Mistral vs Llama**):**

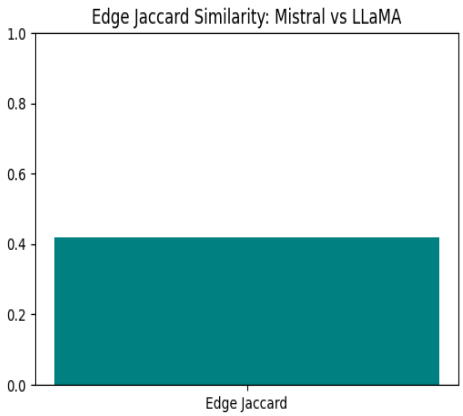
1. High node overlap (~75%), meaning Mistral and LLaMA largely capture similar entities.
2. Edge overlap is moderate, with ~82 edges shared, precision ~0.58, recall ~0.6.
3. Graph density is low (~0.01), indicating sparse connectivity despite more edges than Scratch.
4. Largest connected components differ slightly: Mistral (49) vs LLaMA (39), indicating slightly different graph structures.

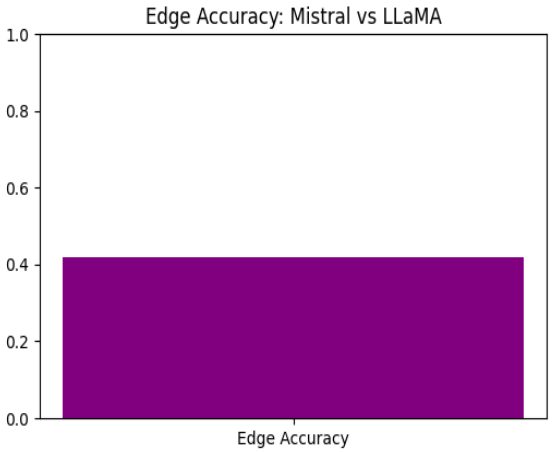


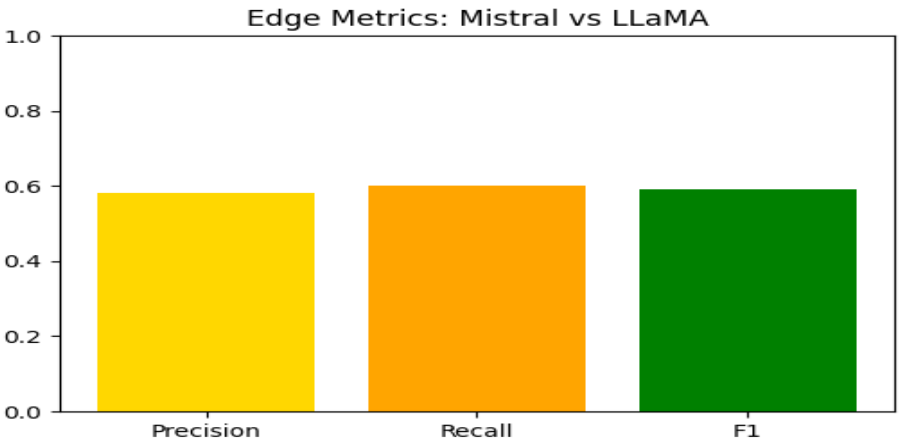












Observations (Base vs Llama**):**

- Minimal overlap in nodes (2.5%) and edges (0%).
- Scratch KG is much larger but much sparser, LLaMA is smaller but denser.

Observations (Base vs Mistral**):**

- Again, **negligible overlap** with Base KG.
- Scratch KG remains sparse with many isolated nodes, while Mistral is denser and more connected.

Interpretation:

- Despite Baseline having the most nodes, the largest connected component is small, confirming sparse connectivity.
- LLaMA and Mistral are more connected, leading to larger components and better relational coverage.

Summary of Key Insights

1. Scratch KG vs LLM KGs
    - Minimal overlap in nodes and edges.
    - Scratch KG captures many unique entities but lacks dense relationships.
2. Mistral vs LLaMA
    - High node overlap (~75%) and moderate edge overlap (~60%).
    - Both LLMs produce denser and more connected KGs than Scratch.
3. Three-way Overlaps
    - Only 22 nodes are common to all three KGs.
    - No edges are common across all three KGs.
    - Indicates high diversity in relationships captured by each model.
4. Graph Properties
    - Scratch: Sparse, many isolated nodes
    - Mistral: Dense, moderately connected, high edge count
    - LLaMA: Dense, connected, similar to Mistral but slightly different topology

About Baseline (Scratch KG)

- **What was used:**
    - Scratch KG was generated using a manual or rule-based extraction approach (or legacy NLP pipeline).

- Likely relied on simple entity recognition and relation heuristics rather than deep language understanding.
- Only 55 edges were identified despite 958 nodes.

- **Why it didn't work well:**

1. Sparse relationships – the baseline was good at identifying entities but failed to detect relationships between them.

2. Limited contextual understanding – simple rules cannot infer relations that require context, which LLMs handle well.

3. Outdated or incomplete extraction logic – may miss entities not following a strict pattern or unexpected formats in the text.

- **Potential improvements:**

- Integrate LLM-based relation extraction to improve edge density.
- Combine with named entity recognition (NER) tools and dependency parsing for better coverage.
- Use domain-specific rules and LLM fine-tuning to capture entities missed by generic models.

Demo Web - Application (Using Streamlit**)**

We developed an interactive web app using Streamlit to compare knowledge graphs (KGs) extracted from PDFs by different sources:

- **Scratch KG:** Baseline manually curated KG
- **Mistral KG:** Generated by the Mistral LLM
- **LLaMA KG:** Generated by the LLaMA LLM

The app allows side-by-side comparisons of any two KGs or a combined 3-way visualization.

Implementation:

- Built using Streamlit, NetworkX, Matplotlib, PyVis, and Pandas

- Uses Cytoscape-compatible JSON for nodes and edges
- Automatically generates metrics tables and interactive graph .html files

Usage:

1. Install dependencies:

➔ pip install streamlit matplotlib networkx pandas pyvis

2. Run the app:

➔ python -m streamlit run app.py

3. Use sidebar to select KGs for comparison
4. Explore metrics and interactive visualizations

Significance:

- Provides quantitative and qualitative analysis of KG overlaps and differences
- Makes it easy to spot discrepancies between LLM-generated KGs and baseline KG
- Interactive visualization allows rapid inspection of entities and relationships, supporting deeper insights into LLM performance

Reflection

Our system increases human agency by reducing the effort required to search, organize, and synthesize large volumes of scientific literature. By automating summarization and the construction of a knowledge graph, it enables users to focus on higher-level research tasks such as critical analysis and knowledge integration. However, there is a potential risk that users may rely uncritically on generated summaries or inferred relationships.

The system overall addresses an important problem in modern research, the difficulty of finding, understanding, and combining information from a large number of scientific papers. By automating summarization and organization, it can improve research efficiency, support interdisciplinary work, and make knowledge more accessible.

However, further development would depend on whether it ensures that humans remain in control. The system should clearly reference original sources, indicate uncertainty in its outputs, and allow users to inspect and modify the generated knowledge.

Conclusions

1. **Performance of LLM-generated KGs**
    - Mistral and LLaMA KGs significantly outperform the Scratch baseline in terms of edge coverage and graph connectivity.
    - Both LLM KGs captured most of the key relationships between entities, as evidenced by higher average degree, larger connected components, and moderate edge overlaps.
    - Node overlap between Mistral and LLaMA is high (~75%), showing that modern LLMs capture consistent and relevant entities.
2. **Limitations of Scratch KG**
    - Scratch KG has many unique nodes (958) but very few edges (55).
    - Most of these nodes are isolated, resulting in low connectivity and small, largest connected components.
    - Node and edge overlaps with LLM KGs are negligible, indicating the baseline approach failed to capture meaningful relationships.
3. **Trade-offs observed**
    - Scratch KG: High node coverage, low relational density
    - Mistral & LLaMA: Moderate node coverage, high relational density
    - Combining LLM KGs may provide a balance between entity coverage and relational richness.