MEETING - 04

* Graphusion - ~~[illegible]~~ Implementation → (✓)
  - → pipeline → (~~[illegible]~~) ✓

Integration with Neo 4j (✓)

⇓

2 options

- Neo 4j Aura DB (online) (✓)
- Neo 4j Docker image (offline)

GRAPHUSION - 03 Steps

① Seed Concepts Extraction

1. NLP techniques
   - Alternative ⇓ Use LLM for it

② Triplet Extraction

→ Pure prompt Engineering

→ Other methods like → fixed functions using (Not recommended) base* NLP

~~[illegible]~~

Alternative

③ Fusion of Graph

to resolve

1. Entity Resolution
2. Conflict Resolution b/w Relationships

Future Tasks

1. Identify a dataset to test RAG GraphRAG pipeline.

2. Integrate KG (Generated by Graphusion) with RAPTOR.

   ⇓

   2 layer Retrieval.

3. Can try different methods of KG Generation & use Graphusion pipeline for
   - → Entity Resolution
   - → Conflict Resolution b/w Relationships.

   Langchain
   OR
   Neo4j
   OR
   Llama Index
   OR
   KAG

4. Integrate Self - RAG (Agentic RAG) to make sure 100% authentic answer or No answer (No room for Hallucination)

   { Langgraph Agents }

Current Approach

→ Fixed ontology for Relationship

For each topic → [(S, O, P) (S, O, P) (S, O, P) (S, O, P)] → Merge ⇓ KG

→ Fusion technique :-

For each (S) or (P) in KG

Prompt Engineering → fix (S) & (P) to avoid duplication

⇓

deduplication

→ Fix relationship degree & direction

~~Next Approach~~

⇓

- CONVERT THE KG into graph Docs to make LANGCHAIN COMPATIBLE.

New Approach

Give freedom to LLM Graph transformer class to generate KG

↓

Good in identifying hidden relationships.

Reason :- → No ontology restrictions.
→ Use actual relationship name mentioned in the original doc

Eg :-

Graphusion

Machine learning → is-a-prerequisite of → Deep learning.

Fixed ontology.

Langchain

Deep learning → Subset → Machine learning

Original sentence :- DL is a subset of ML & ML is a subset of AI.

Use → Fusion technique of Graph Fusion as discussed