WASEDA University
Graduate School of Information, Production and Systems

KGGen: Extracting Knowledge Graphs from Plain Text with Language Models

HUANG JIAHUI

2025/10/16

Data Engineering Lab, IPS, Waseda Univ.

Outline

Outline

- ◆ Introduction
- ◆ Methodology
- ◆ Experiment
- ◆ Conclusion

Introduction

- **Knowledge Graphs (KGs) consist of a set of subject-predicate-object triples**
  - Example: (cryptocurrencies, enhance, security)
- **Most real-world KGs are far from complete**
  - Scarcity and incompleteness
  - Questionable quality
    - Noise info
    - Lack of domain-specific knowledge
    - Hallucinations

Introduction

Existing methods for extracting KGs from plain text

OpenIE

- Generates a “dependency parse” for each sentence
- A learned classifier then traverses each edge in the dependency parse
- Not require its input text to have a specific structure

GraphRAG

- Integrated graph-based knowledge retrieval with language models
- Generating KGs from plain text to use as its database
- Prompting LMs to extract node-entities and relationships
- Aggregates well-connected nodes into communities

Methodology

KGGen: Text-to-Knowledge-Graph

➢ Overview

- An open-source package that uses LMs to extract high-quality KGs from plain text
- A multi-stage approach involving an LLM (in this case, GPT-4o)

➢ 3 Modules

- ➢ **Generate**: Entity and Relation Extraction
- ➢ **Aggregate**: Aggregation
- ➢ **Cluster**: Entity and Edge Clustering

Methodology

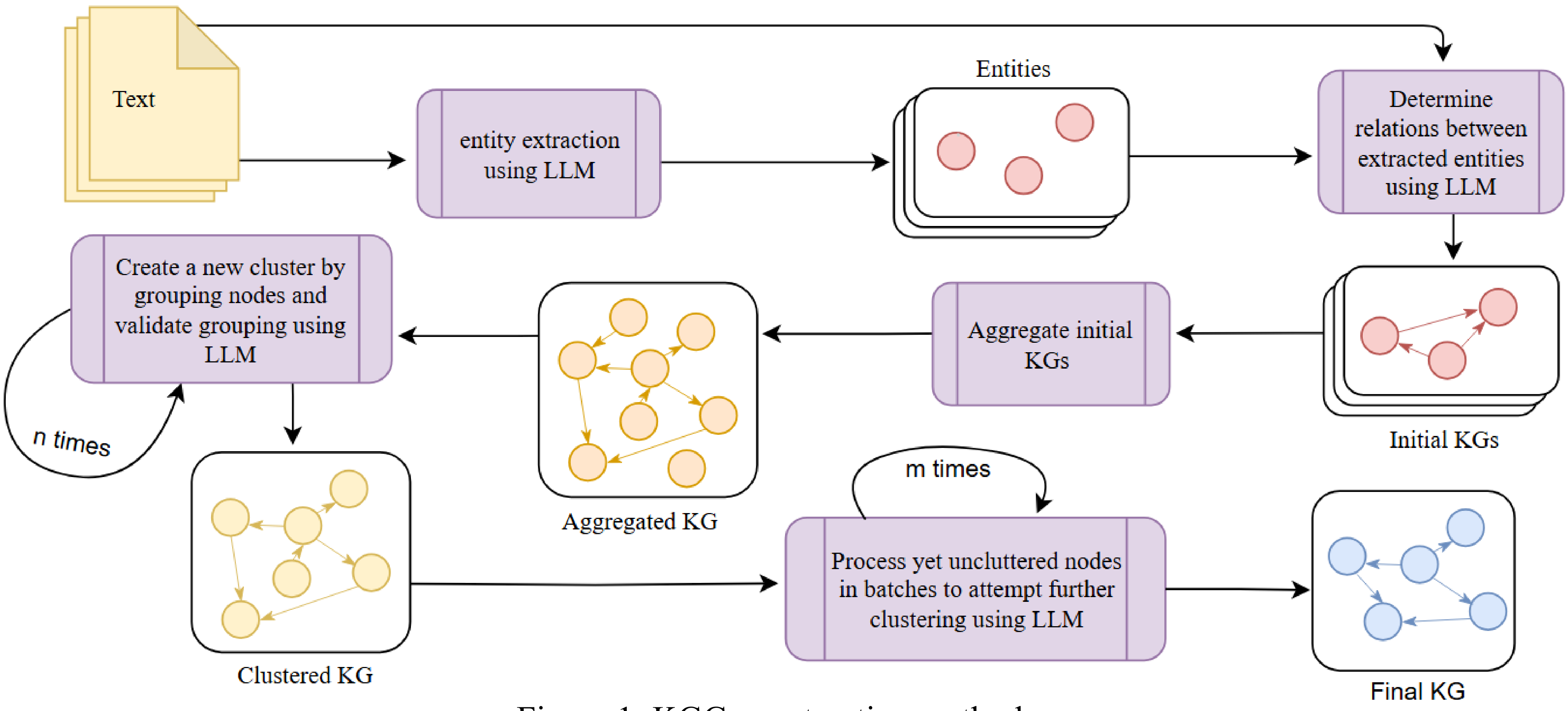
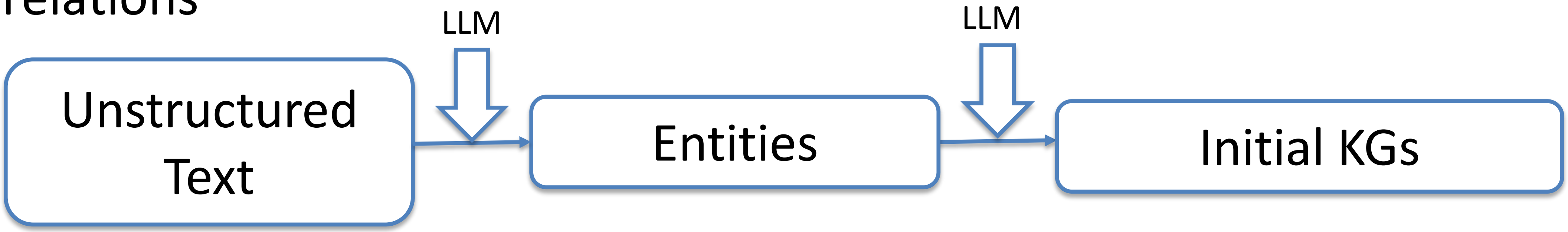

Text
entity extraction using LLM
Entities
Determine relations between extracted entities using LLM
Initial KGs
Aggregate initial KGs
Aggregated KG
Create a new cluster by grouping nodes and validate grouping using LLM
n times
Clustered KG
m times
Process yet uncluttered nodes in batches to attempt further clustering using LLM
Final KG

Figure 1: KGGen extraction method

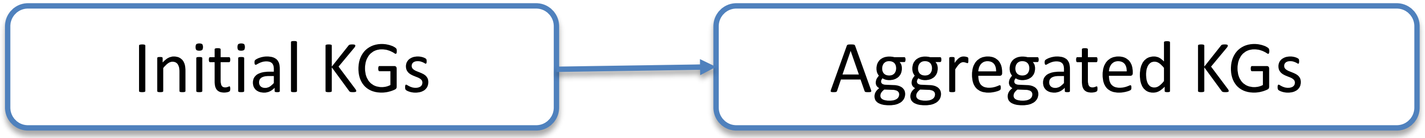
Generate: Entity and Relation Extraction

- **2-step approach**
- **1st step:** takes unstructured text as input and invoke the GPT-4o model to output detected entities
- **2nd step:** invoke a second LLM call to output the subject-predicate-object relations



Aggregate: Aggregation

- Collect all the unique entities and edges across all source graphs
- Combine them into a single graph
  - All entities and edges are normalized to be in lowercase letters only
  - Reduces redundancy



Cluster: Entity and Edge Clustering(key innovation)

- Aggregated KGs is a raw graph containing duplicate or synonymous entities and possibly redundant edges
- Merge nodes and edges representing the same real-world entity or concept
- Use an iterative LLM-based approach to clustering(inspired by humans)
- Use LLM-as-a-Judge call to validate

Cluster: Entity and Edge Clustering(key innovation)

Workflow

1. Propose Cluster: The entire entities list is passed in context to the LLM, and it attempts to extract a single cluster
2. Validate Cluster: LLM-as-a-Judge gives Yes/No on semantic consistency
3. Label Cluster: Assign a representative name capturing shared meaning
4. Iterate: Repeat until no new valid clusters found
5. Batch Process: Check remaining entities with batches size b
6. Re-validate: Re-run LLM-as-a-Judge check for any updated cluster
7. Stop when all entities are clustered

- e.g. “vulnerabilities”, “vulnerable”, and “weaknesses”.

Benchmark: MINE

- Measure of Information in Nodes and Edges
- The first benchmark that measures a knowledge-graph extractor’s ability to capture and distill a body of text into a KG

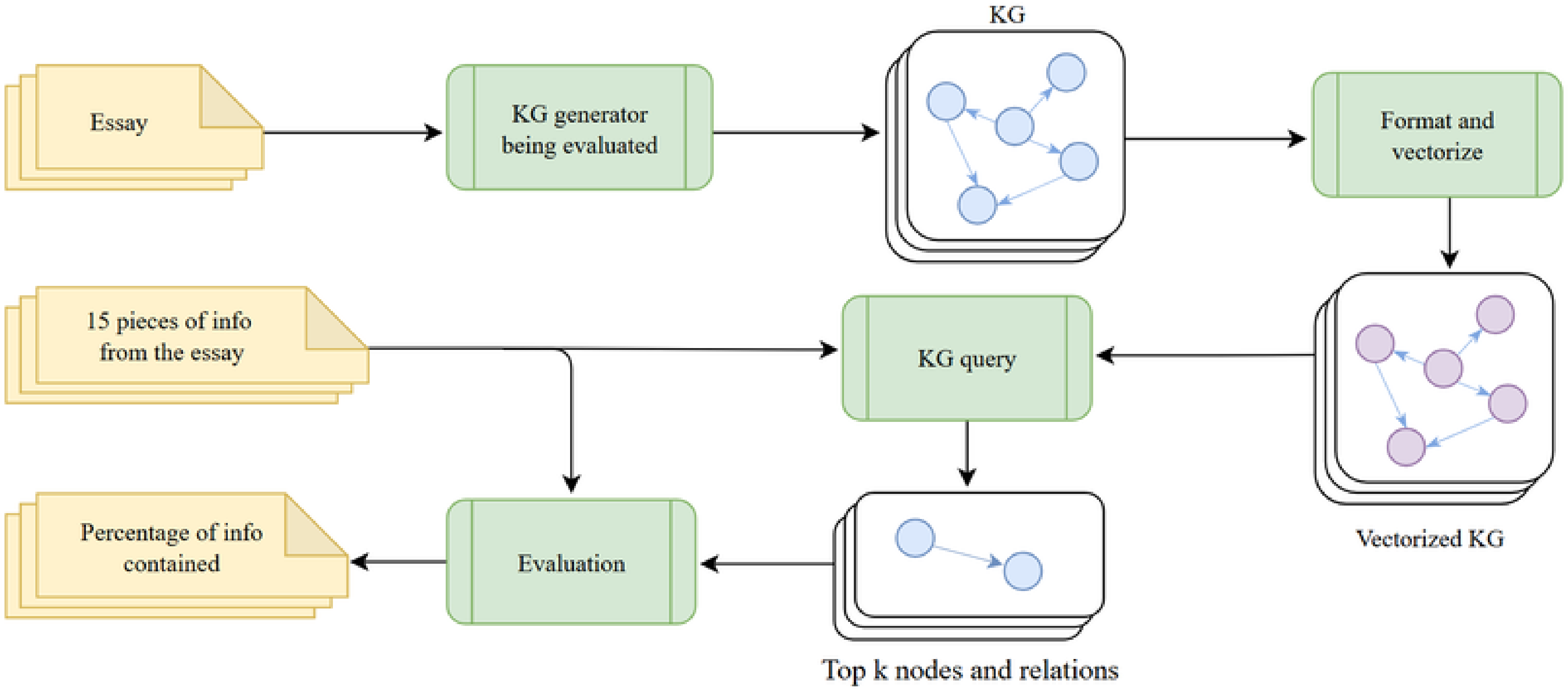

KG
Essay
KG generator being evaluated
Format and vectorize
15 pieces of info from the essay
KG query
Vectorized KG
Percentage of info contained
Evaluation
Top k nodes and relations

Figure 2: Evaluation process used in MINE

Experiment

◆ Result

KGGen	GraphRAG	OpenIE
66.07%	47.8%	29.84%

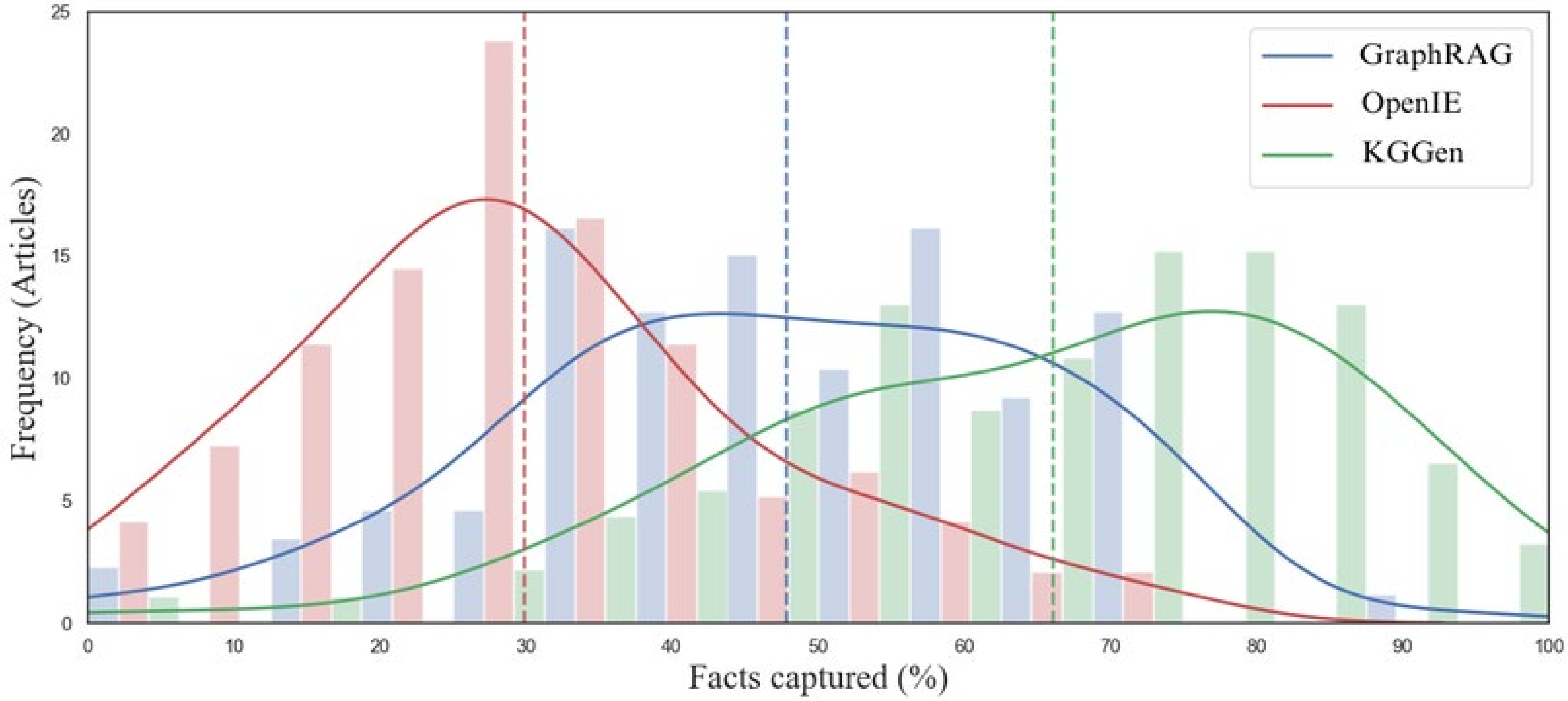
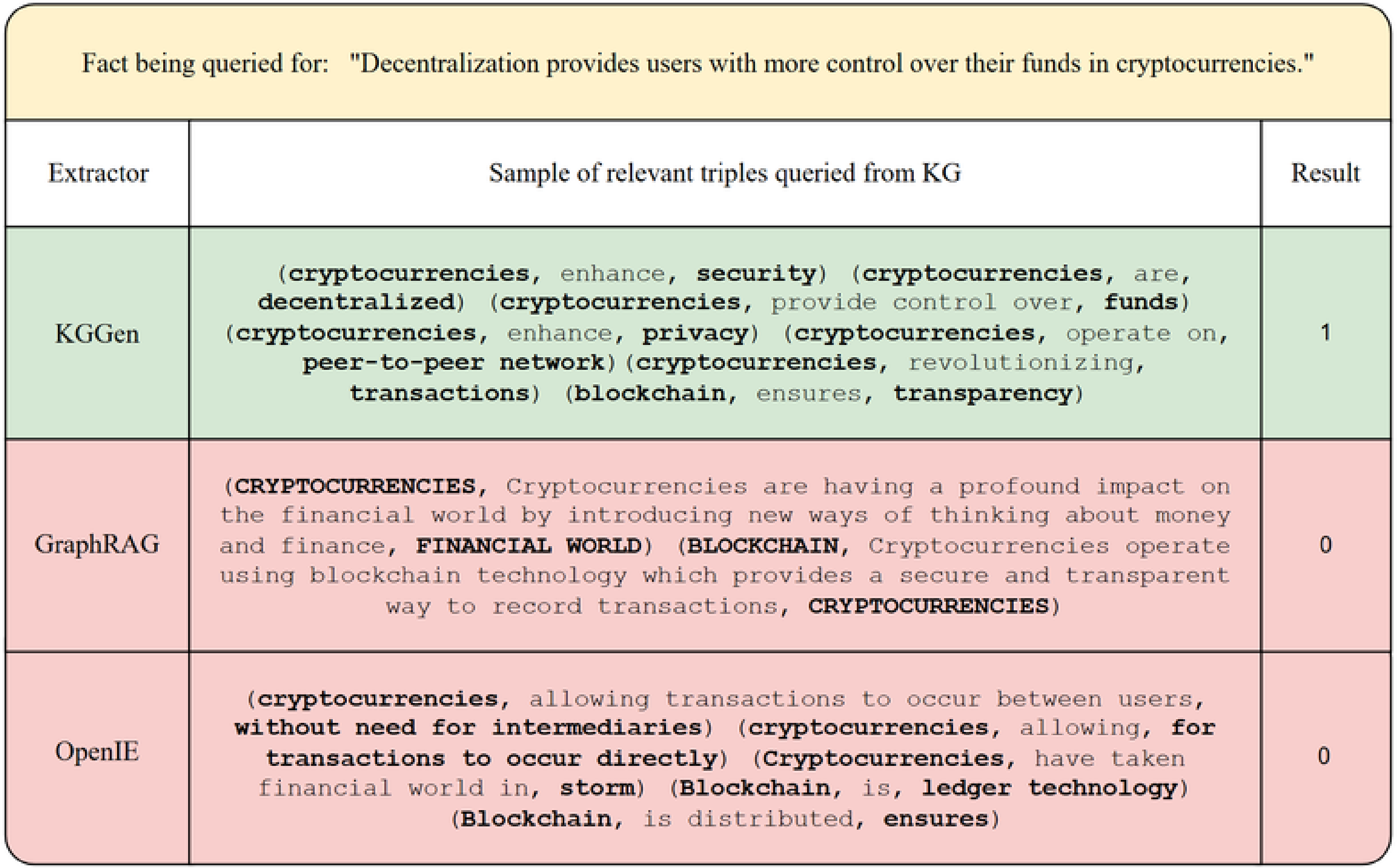


Figure 3: Distribution of MINE scores across 100 articles for GraphRAG, OpenIE, and KGGen

Experiment



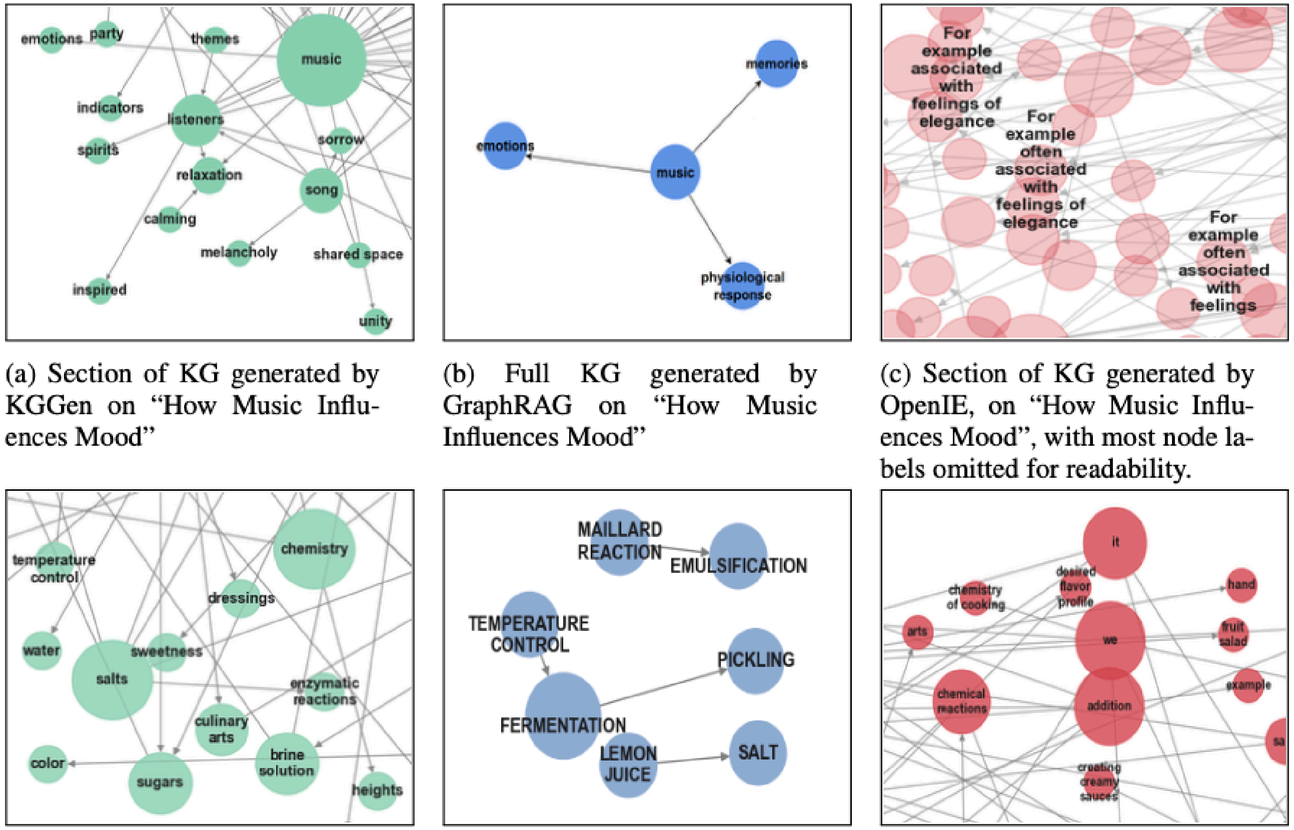
Fact being queried for: "Decentralization provides users with more control over their funds in cryptocurrencies."

Extractor	Sample of relevant triples queried from KG	Result
KGGen	(**cryptocurrencies**, enhance, **security**) (**cryptocurrencies**, are, **decentralized**) (**cryptocurrencies**, provide control over, **funds**) (**cryptocurrencies**, enhance, **privacy**) (**cryptocurrencies**, operate on, **peer-to-peer network**) (**cryptocurrencies**, revolutionizing, **transactions**) (**blockchain**, ensures, **transparency**)	1
GraphRAG	(**CRYPTOCURRENCIES**, Cryptocurrencies are having a profound impact on the financial world by introducing new ways of thinking about money and finance, **FINANCIAL WORLD**) (**BLOCKCHAIN**, Cryptocurrencies operate using blockchain technology which provides a secure and transparent way to record transactions, **CRYPTOCURRENCIES**)	0
OpenIE	(**cryptocurrencies**, allowing transactions to occur between users, **without need for intermediaries**) (**cryptocurrencies**, allowing, **for transactions to occur directly**) (**Cryptocurrencies**, have taken financial world in, **storm**) (**Blockchain**, is, **ledger technology**) (**Blockchain**, is distributed, **ensures**)	0

Figure 4: An example query from the MINE benchmark, along with relevant relations in the KGs extracted by KGGen, GraphRAG, and OpenIE

Experiment

- **GraphRAG**
  - Generates a minimal number of nodes and connections
  - Omission of critical relationships and information
- **OpenIE**
  - Redundant and generic nodes
- **KGGen**
  - KGs are dense and coherent
  - Capturing critical relationships and information



(a) Section of KG generated by KGGen on "How Music Influences Mood"

(b) Full KG generated by GraphRAG on "How Music Influences Mood"

(c) Section of KG generated by OpenIE, on "How Music Influences Mood", with most node labels omitted for readability.

(d) Section of KG generated by KGGen on "The Chemistry of Cooking"

(e) Full KG generated by GraphRAG on "The Chemistry of Cooking"

(f) Section of KG generated by OpenIE on "The Chemistry of Cooking"

Figure 5: Visual comparison of KGs generated using KGGen, GraphRAG, and OpenIE. Results show that KGGen discovers more informative nodes to estimate a richer graph compared to GraphRAG, and collapses synonyms to discover a more informative graph than OpenIE.

Conclusion

➢ KGGen

- KGGen integrates three complementary processes (Generate → Aggregate → Cluster) that collectively enhance both precision and semantic density

➢ LLM-as-a-Judge

- Creatively use LLMs to evaluate and refine the KG
- This self-evaluation mechanism mimics human-level semantic judgment

➢ Benchmark: MINE

- The first benchmark that measures a knowledge-graph extractor’s ability to capture and distill a body of text into a KG

Future Work

- **Limitation**
  - KGGen: Over or under-clustering
  - MINE: Short corpora
- **Future Work**
  - Explore more advanced clustering mechanisms to improve the quality of KGs
  - Extend to larger corpora in the future to verify its practicality in constructing real large-scale knowledge graphs

Reference

[1] Mo, Belinda, et al. "KGGen: Extracting Knowledge Graphs from Plain Text with Language Models." arXiv preprint arXiv:2502.09956 (2025).

[2] Huang, Haoyu, et al. "Can LLMs be Good Graph Judge for Knowledge Graph Construction?." arXiv preprint arXiv:2411.17388 (2024).

[3] Anuyah, Sydney, et al. "Automated Knowledge Graph Construction using Large Language Models and Sentence Complexity Modelling." arXiv preprint arXiv:2509.17289 (2025).

Thank You