KGGen: Extracting Knowledge Graphs from Plain Text with Language Models

Belinda Mo*[1], **Kyssen Yu***[2], **Joshua Kazdan***[1],
Joan Cabezas[3], **Proud Mpala**[1], **Lisa Yu**[2],
Chris Cundy[4], **Charilaos Kanatsoulis**[1], **Sanmi Koyejo**[1]
[1]Stanford University [2]University of Toronto [3]Independent [4] FAR AI

Abstract

Recent interest in building foundation models for knowledge graphs has highlighted a fundamental challenge: knowledge graph data is scarce. The best-known knowledge graphs are primarily human-labeled, created by pattern-matching, or extracted using early NLP techniques. While human-generated knowledge graphs are in short supply, automatically extracted ones are of questionable quality. We present KGGen, a novel text-to-knowledge-graph generator that uses language models to extract high-quality graphs from plain text with a novel entity resolution approach that clusters related entities, significantly reducing the sparsity problem that plagues existing extractors. Unlike other KG generators, KGGen clusters and de-duplicates related entities to reduce sparsity in extracted KGs. Along with KGGen, we release Measure of Information in Nodes and Edges (MINE), the first benchmark to test an extractor's ability to produce a useful KG from plain text. We benchmark our new tool against leading existing generators such as Microsoft's GraphRAG; we achieve comparable retrieval accuracy on the generated graphs and better information retention. Moreover, our graphs exhibit more concise and generalizable entities and relations. Our code is open-sourced at https://github.com/stair-lab/kg-gen/.

1 Introduction

Knowledge graph (KG) applications and graph-based Retrieval-Augmented Generation (RAG) systems are increasingly bottlenecked by the scarcity and incompleteness of available KGs. KGs consist of a set of subject-predicate-object triples, and have become a fundamental data structure for information retrieval [Schneider, 1973]. Most real-world KGs, including Wikidata [contributors, 2024], DBpedia [Lehmann et al., 2015], and YAGO [Suchanek et al., 2007], are far from complete, with many missing relations between entities [Shenoy et al., 2021]. The lack of domain-specific and verified graph data poses a serious challenge for downstream tasks such as KG embeddings, graph RAG, and synthetic graph training data.

Several synthetic plain-text-to-KG extractors have been proposed to address KG scarcity, most prominently including OpenIE [Angeli et al., 2015] and Microsoft's GraphRAG [Larson and Truitt, 2024]. Both OpenIE and GraphRAG extract entities and relations directly from text, but they lack effective mechanisms for entity resolution and relation normalization. This leads to graphs with nearly as many unique relation types as edges, resulting in sparse, disconnected knowledge representations that limit their utility for downstream tasks. To solve this problem, we propose KGGen, a text-to-knowledge-graph generator that leverages language models (LMs) and an algorithm for entity and edge resolution to extract high-quality, dense KGs from text. First, KGGen uses an LM-based extractor to read unstructured text and predict subject-predicate-object triples to capture entities and relations; after extracting the triples, it applies a novel, iterative clustering algorithm to refine the

*Equal Contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

raw graph. Inspired by crowd-sourcing strategies for entity resolution [Wang et al., 2012], KGGen identifies nodes that refer to the same underlying entities, and consolidates edges that have equivalent meanings.

The nascent field of plain-text-to-knowledge graph extraction currently lacks benchmarks to measure the fidelity of KG generation from text. To close this gap, we provide two new benchmarks: the first captures information retention from short texts; the second, based on WikiQA, measures knowledge retrieval capabilities for graphs generated from multi-million token, web-based knowledge databases. On these benchmarks, KGGen performs comparably to GraphRAG. However, KGGen exhibits far better scaling properties with respect to information compression and graph sparsity as the plain-text database length increases.

To summarize our contributions:

1. We introduce KGGen, an open-source package that uses LMs to extract high-quality KGs from plain text. Our package is available as a Python library.
2. We develop benchmarks to drive improvements in plain-text-to-knowledge-graph extraction, and measure KGGen's performance on these benchmarks.
3. We show that KGGen exhibits improved scaling with respect to the size of the text source relative to past methods.

2 Related Work

Interest in automated methods to produce structured text to store ontologies dates back to at least 2001 when large volumes of plain text began to flood the fledgling internet [Maedche and Staab, 2001]. KG extraction from unstructured text has seen significant advances through rule-based and LM-powered approaches in the last 15 years. Early work [Suchanek et al., 2007] used hard-coded rules to develop YAGO, a KG extracted from Wikipedia containing over five million facts, and rules-based extraction still has appeal for those producing KGs in multi-modal domains today [Norabid and Fauzi, 2022, Oramas et al., 2015]. With the development of modern natural language processing, hard-coded rules generally ceded to more advanced approaches based on neural networks. For instance, OpenIE [Angeli et al., 2015] provides a two-tiered extraction system: first, self-contained clauses are identified by a classifier; then, [Angeli et al., 2015] run natural logic inference to extract the most representative entities and relations from the identified clauses. Stanford KBP [Angeli et al., 2013] presents another seminal early approach to using deep networks for entity extraction.

As early as 2015, some hypothesized that extracting KGs would go hand-in-hand with developing better language models [Domeniconi et al., 2015]. More recently, evidence has emerged that transformer-based architectures can identify complex relationships between entities, leading to a wave of transformer-based KG extraction techniques, which range from fully automatic [Qiao et al., 2022, Arsenyan et al., 2023, Zhang and Soh, 2024] to human-assisted [Kommineni et al., 2024]. Our contribution to the extraction literature is to build KGs conducive to embedding algorithms such as TransE and TransR [Bordes et al., 2013, Lin et al., 2015]. We observed that when one extracts KGs from plaintext, the nodes and relations are often so specific that they are unique. This causes the estimation of embeddings to be under-specified. We develop a method for automatic KG extraction from plain text that clusters similar nodes and edges to prevent this under-specification. This leads to a KG with better connectivity and more functional nodes and edges.

Evaluating the quality of knowledge graphs is important to ensure usefulness and reliability in downstream applications. Early evaluation methods focused primarily on directly assessing aspects such as completeness and connectivity or using rule-based statistical methods, while recent approaches emphasize usability in downstream applications and incorporation of semantic coherence[Xue and Zou, 2023].

In the late 2000s, research focused on assessing the correctness and consistency of KGs. The evaluations relied on expert annotations by selecting random facts from the generated KG and then calculating the accuracy of those facts. [Suchanek et al., 2007] This proved to be laborious and prone to errors. This led to accuracy approximation methods like KGEval [Ojha and Talukdar, 2017] and Two-State Weight Clustering Sampling(TWCS) [Gao et al., 2018], which employed sampling methods with statistical guarantees as well as use less annotation labor. As the KGs became larger and more diverse, particularly with the rise of automated extraction techniques from web data, this generated more pressure on annotators, leading to methods like Monte-Carlo search being used for the interactive annotation of triples [Qi et al., 2022]. Furthermore, because accuracy alone did not fully capture the complexity of the knowledge graph, more evaluation metrics like completeness were used to characterize the quality of knowledge graphs. [Issa et al., 2021].

In recent years, the evaluation of knowledge graphs (KGs) has increasingly focused on their role in downstream AI applications, such as augmenting language models [Schneider et al., 2022] and recommendation systems [He et al., 2020]. As a result, semantic coherence and usability have become key criteria for assessing the quality of extracted knowledge graphs.

Two notable approaches to KG evaluation are the LP-Measure and the triple trustworthiness measurement (KGTtm) model. LP-Measure assesses the quality of a KG through link prediction tasks, eliminating the need for human labor or a gold standard [Zhu et al., 2023]. This method evaluates KGs based on their consistency and redundancy by removing a portion of the graph and testing whether the removed triples can be recovered through link prediction tools. Empirical evidence suggests that LP-Measure can effectively distinguish between “good" and “bad" KGs. The KGTtm model, on the other hand, evaluates the coherence of triples within a knowledge graph Jia et al. [2019]. Based on these evaluation methods, frameworks like Knowledge Graph Evaluation via Downstream Tasks(KGrEaT) and DiffQ(differential testing) emerged. KGrEaT provides a comprehensive assessment of KGs by evaluating their performance on downstream tasks such as classification, clustering, and recommendation [Heist et al., 2023] rather than focusing solely on correctness or completeness. In contrast, DiffQ uses embedding models to evaluate the KG’s quality and assign a DiffQ Score, resulting in improved KG quality assessment Tan et al. [2024].

3 Existing Methods

Before describing KGGen, we explain two popular existing methods for extracting KGs from plain text, which will serve as a basis for comparison throughout the rest of this paper.

3.1 OpenIE

Open Information Extraction (OpenIE) was implemented by Stanford CoreNLP based on Angeli et al. [2015]. It first generates a “dependency parse" for each sentence using the Stanford CoreNLP pipeline. A trained classifier then traverses each edge in the dependency parse, deciding whether to create, continue, or stop processing a clause. These decisions split complex sentences into shorter, self-contained clauses. From these clauses, the system produces (*subject, relation, object*) tuples, each accompanied by a confidence score. Because OpenIE does not require its input text to have a specific structure, OpenIE can handle text in any format.

3.2 GraphRAG

GraphRAG, developed by Microsoft in 2024, integrates graph-based knowledge retrieval with language models (LMs) [Larson and Truitt, 2024]. As a first step, GraphRAG provides functionality for generating KGs from plain text, which serve as its database for retrieval. In this process, GraphRAG creates a graph by prompting LMs to extract node-entities and relationships between these entities. Throughout this extraction, few-shot prompting provides the LM with examples of desireable extractions. GraphRAG aggregates well-connected nodes into communities and generates a summary for each community. The final graph consists of the nodes and their relationships along with communities their summaries.

4 KGGen: Knowledge Graphs From Plain Text

Unlike most previous methods of LLM-based KG extraction, we rely on a multi-stage approach (1) extract entity and relations from each source text using an LLM, (2) aggregate graphs across sources, and (3) iteratively resolve duplicate entities and edges using a hybrid of LLM and traditional informational retrieval methods.

We impose strong constraints on the LLM via prompting to prevent it from incorrectly grouping together entities or edges that are similar in meaning but not actually the same - for example, conflating "Type 1 diabetes" and "Type 2 diabetes," "hypertension" and "stress," or "MRI" and "CT scan.". We introduce multiple passes through our extracted edges and edges to resolve similar entities and consolidate the number of edge types. Entity and edge resolution prevents the formation of sparse KGs, which may produce meaningless KG embeddings under standard algorithms such as TransE.

Our extraction method involves several steps, which we outline below. The exact prompts for each step can be found in Appendix A, and the process is illustrated in Figure 1.

4.1 Entity and Relation Extraction

The first stage takes unstructured text as input and produces an initial knowledge graph as extracted triples. We use Google's Gemini 2.0 Flash as the language model to provide structured output via DSPy signatures. The first step takes in source text and extracts a list of entities. Given the source text and entities list, the second step outputs a list of subject-predicate-object relations. Each step corresponds to a DSPy signature that specifies instructions for the LLM to follow in its docstring. We find this 2-step approach works better to ensure consistency between entities.

4.2 Aggregation

After extracting triples from each source text, we collect all the unique entities and edges across all source graphs and combine them into a single graph. All entities and edges are normalized to be in lowercase letters only. The aggregation step reduces redundancy in the KG. Note that the aggregation step does not require an LLM.

4.3 Entity and Edge Resolution

After extraction and aggregation, we typically have a raw graph containing duplicate or synonymous entities and possibly redundant edges. The resolution stage is a key innovation in our KG extraction methodology that merges nodes and edges representing the same real-world entity or concept. Our resolution process employs a two-stage approach combining embedding-based clustering with LLM-based de-duplication to efficiently handle large knowledge graphs. The approach is applied to both entity and edge items separately:

First, all items in the graph are clustered. We get the semantic embeddings of every item using S-BERT and cluster using k-means into clusters of 128 items.

1. For each item in a cluster, we retrieve the top-k most semantically similar items, where k=16, using a fused BM25 and semantic embedding approach.
2. Then, the LLM is prompted to identify exact duplicates from this set, considering variations in tense, plurality, case, abbreviations, and shorthand forms.
3. For each set of duplicates, the LLM selects a canonical representative that best captures the shared meaning, similar to aliases that Wikidata uses. Cluster maps track which entities belong to which alias.
4. The item and its duplicates are removed from the cluster and steps 1-3 repeat until no items remain in the cluster.

This approach enables effective de-duplication even for very large knowledge graphs by processing semantic clusters in parallel. When processing our largest 20M chunk dataset, this method successfully consolidated entities like "Olympic Winter Games", "Winter Olympics", and "winter Olympic games" into a single canonical representation.

5 Benchmarks for Extraction Performance

Although a handful of existing methods attempt to extract KGs from plain text, it is difficult to measure progress on this task due to the lack of existing benchmarks. As a remedy, we produce the Measure of Information in Nodes and Edges (MINE), the first benchmark that directly measures a

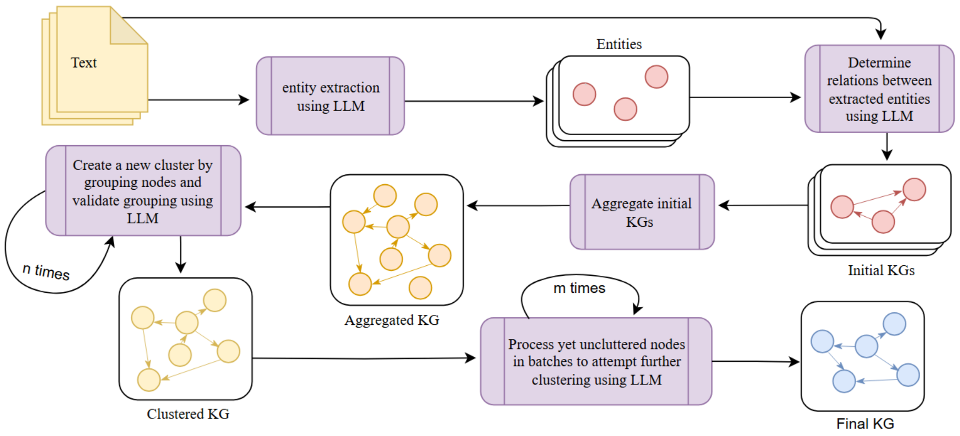

Text
entity extraction using LLM
Entities
Determine relations between extracted entities using LLM
Initial KGs
Aggregate initial KGs
Aggregated KG
Create a new cluster by grouping nodes and validate grouping using LLM
n times
Clustered KG
m times
Process yet uncluttered nodes in batches to attempt further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

Figure 1: KGGen extraction method

knowledge-graph extractor's ability to capture and distill a body of text into a KG. MINE contains two tasks: the first assesses how well a KG extractor captures the information in short, page-length text; the second measures downstream RAG performance on multi-million token datasets. We call these tasks MINE-1 and MINE-2 respectively. MINE-1 ensures that a KG extractor accurately represents the source text, and MINE-2 gauges the practicality of the knowledge graphs for realistic applications.

5.1 MINE-1: Knowledge Retention

MINE-1 approximates the fraction of information a KG extractor is able to capture from an article without relying on downstream tasks, which can obscure whether performance gains stem from the KG extractor itself or from aspects of the extraction process.

MINE-1 consists of 100 articles, each accompanied by 15 facts that are known to be present in the article. The dataset has the following characteristics: articles have a mean length of 592 words (std. 85 words, range: 440-976 words) and cover diverse topics including Arts, Culture & Society (24 articles), Science (27 articles), Technology (19 articles), Psychology/Human Experience (18 articles), and History & Civilization (17 articles). Articles are generated by an LLM to ensure balanced coverage across these domains. For each article, MINE-1 generates a corresponding KG using the extractor being evaluated.

To assess the quality of these KGs, we extract 15 facts from each article using an LLM-based extraction prompt found in Appendix B. We manually verify that the 15 facts are accurate and contained in the article. To measure performance on MINE-1, the KG extractor first extracts a KG from each article. Then, MINE-1 contains a process to verify whether each fact can be recovered from the corresponding KG.

Verification occurs via a semantic query process: both the 15 facts and all KG nodes are embedded using the all-MiniLM-L6-v2 model from SentenceTransformers. For each fact, the verifier retrieves the top-k most semantically similar nodes in the KG, then expands the result to include all nodes within two relations of those top-k nodes, reflecting the fact that KG's are often used for multi-hop reasoning tasks. The subgraph induced by these nodes is passed to an LLM, which is prompted to output a binary score: 1 if the fact can be inferred from the retrieved nodes and relations alone, and 0 otherwise. The prompt is detailed in Appendix B. The final MINE-1 score for a KG extractor is the percentage of the 15 facts scored as 1, averaged across all 100 articles. While LLM-based evaluation introduces potential biases, we validated its reliability by manually scoring 60 randomly selected fact-KG pairs and comparing them to LLM judgments, achieving 90.2% agreement and a correlation of 0.80. The full evaluation pipeline is illustrated in Figure 2.

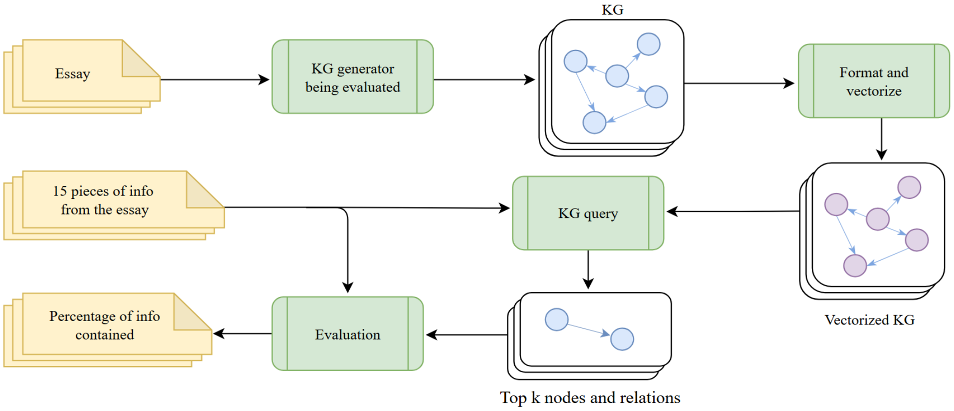

KG
Essay
KG generator being evaluated
Format and vectorize
15 pieces of info from the essay
KG query
Vectorized KG
Percentage of info contained
Evaluation
Top k nodes and relations

Figure 2: Evaluation process used in MINE-1

5.2 MINE-2: KG-Assisted RAG Description

The RAG evaluation is based on the WikiQA dataset Yang et al. [2015], which contains 20,400 questions based on 1,995 Wikipedia articles. Using the method under evaluation, we construct a single KG that aggregates information from all articles referenced in WikiQA. For each question in the dataset, we retrieve the top 10 most relevant triples by embedding both the question and all KG triples using the all-MiniLM-L6-v2 model from SentenceTransformers. We then compute the cosine similarity between the question and each triple, alongside a relevance score produced by BM25. The final similarity score is obtained by combining BM25 relevance score and the cosine similarity score, weighted equally. The 10 triples with the highest combined scores are selected, and we further expand this set by appending 10 additional triples that fall within two hops of the nodes in the top 10 triples to enable multi-hop reasoning.

Since each relation is linked to a source text chunk during generation by KGGen and GraphRAG, we provide the full set of 20 retrieved triples, their associated text chunks, and the original question to an LM, which synthesizes an answer based on these inputs. The complete prompt used can be found in Appendix B. Finally, the LM responses are evaluated using LLM-as-a-Judge to determine whether they contain the correct answer to the question. The prompt used for this final verification step is also included in Appendix B. OpenIE is excluded from this comparison, as it cannot produce KGs that link relations to the original text chunks.

6 Results

We use MINE to benchmark KGGen against leading existing methods of plain-text-to-KG extraction: OpenIE Angeli et al. [2015] and GraphRAG Larson and Truitt [2024]. After providing this quantitative comparison of extraction fidelity, we present qualitative results that demonstrate the advantages of KGGen over past methods.

6.1 Evaluations on MINE-1

Figure 3 displays accuracies from KGGen, OpenIE, and GraphRAG on MINE. Note that KGGen outperforms competing methods. Figure 4 shows an example query from MINE-1 and relevant relations extracted by KGGen, OpenIE, and GraphRAG.

6.1.1 Generalization Across Language Models

To evaluate KGGen's robustness across different foundation models, we tested its performance on MINE-1 using multiple state-of-the-art LLMs. Table 1 shows that KGGen maintains strong performance across different models, with Claude Sonnet 3.5 achieving the highest score of 73%.

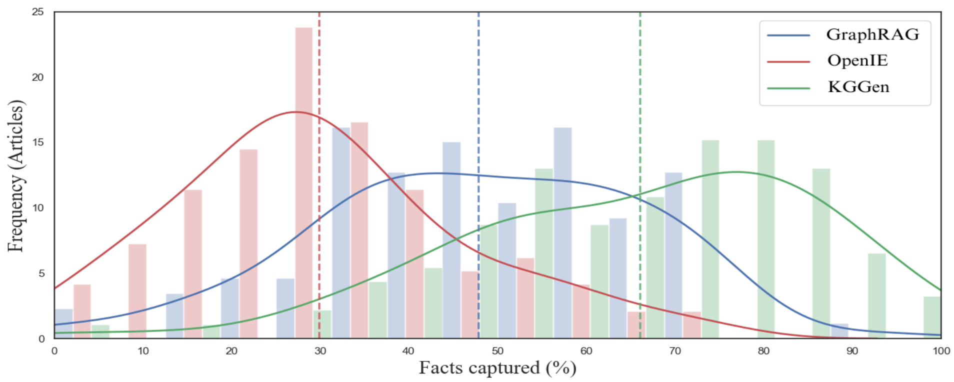
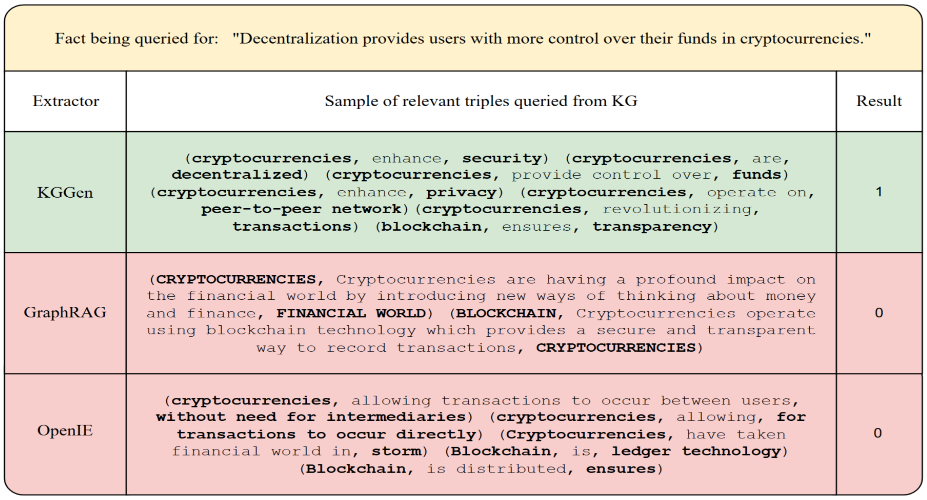


Figure 3: Distribution of MINE-1 scores across 100 articles for GraphRAG, OpenIE, and KGGen. Dotted vertical lines show average performance. KGGen scored 66.07% on average, significantly outperforming GraphRag 47.80% and OpenIE 29.84%.



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-1 benchmark, along with relevant relations in the KGs extracted by KGGen, GraphRAG, and OpenIE. Note that the relation triples extracted by KGGen contain the fact being queried for, whereas the KGs extracted by GraphRAG and OpenIE do not. The relation types extracted by KGGen are more concise and generalize more easily than those from GraphRAG and OpenIE. The full article that these relations were extracted from can be found in Appendix C.

Table 1: Performance comparison of KGGen

(a) KGGen performance on MINE-1 across different language models

Model	MINE-1 Score (%)
Claude Sonnet 3.5	73
GPT-4o	66
Gemini 2.0 Flash	44

(b) Validity of extracted triples across different methods

Method	Valid Triples (%)
KGGen	98/100 (98%)
GraphRAG	0/100 (0%)
OpenIE	55/100 (55%)

KGGen's extraction methodology generalizes well across different foundation models. Although Claude Sonnet 3.5 achieves the highest score of 73%, all tested models maintain reasonable extraction quality, making KGGen adaptable to users' preferred LLM providers.

6.2 Evaluations on MINE-2: RAG performance

Figure 5 shows comparable performance between KGGen and GraphRag on MINE-2.

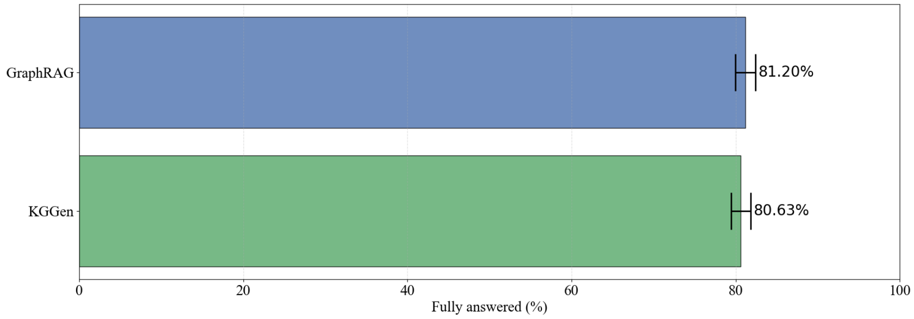


Figure 5: Comparison between KGGen and GraphRag on MINE-2. The two methods perform comparably.

6.3 Evaluation on Human-Annotated Data: SemEval-2010

To evaluate KGGen's extraction quality against human-annotated ground truth, we conducted experiments using the SemEval-2010 Task 8 dataset. We randomly selected 100 sentences from the dataset, each containing two manually labeled target entities. After removing entity markup tags to ensure unbiased extraction, we applied KGGen to extract entities and relationships from each sentence.

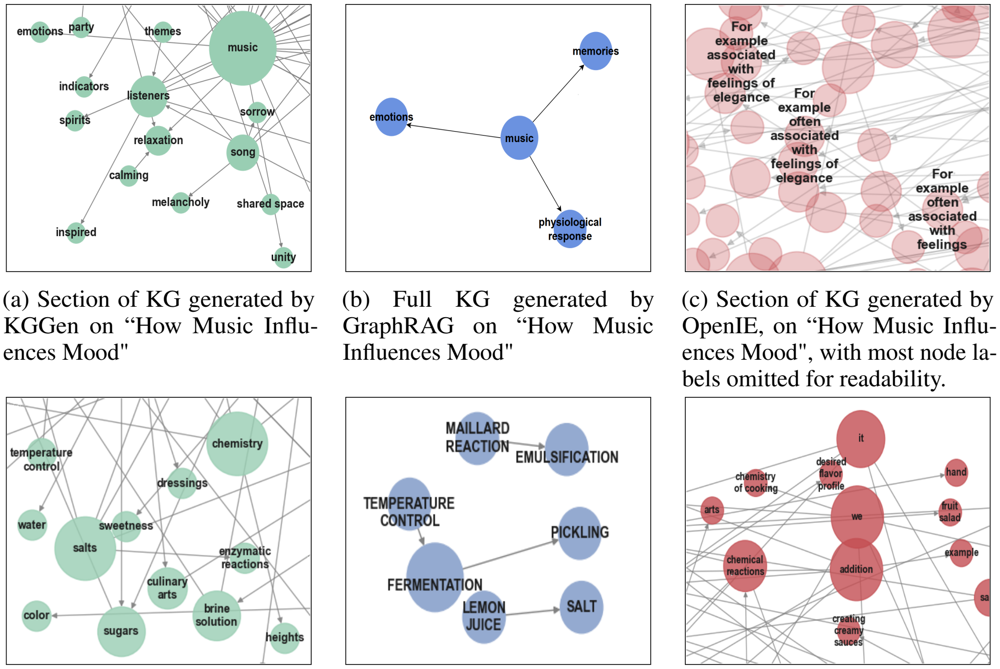
Our evaluation focused on entity extraction accuracy, as the dataset's relation labels consist of only 5 broad categorical types rather than specific semantic relations. We assessed whether both human-annotated target entities appeared in KGGen's extracted entities, allowing for more specific entity descriptions (e.g., "Eurasia exhibition" instead of just "exhibition") that still referred to the same object.

Results show that KGGen successfully captured both target entities in 96% of cases (96/100). The method consistently extracted more detailed entity descriptions compared to human annotations, often identifying additional relevant entities in longer sentences. For example, given the sentence "The ambitious Eurasia exhibition arose from an idea by Achille Bonito Oliva," with target entities 'exhibition' and 'idea', KGGen extracted ['Eurasia exhibition', 'idea', 'Achille Bonito Oliva'], providing more specific and complete entity identification.

6.4 Qualitative Results

We first evaluated the fundamental quality of extracted knowledge graphs by examining whether the extracted triples conform to the basic definition of a knowledge graph: subject-predicate-object triples where subjects and objects are entities (nodes) and predicates are relationships (edges). We randomly selected 100 triples from each method and manually evaluated their validity. The results are shown in Table 1.

Despite GraphRAG's comparable performance on downstream tasks, it does not extract structures that closely resemble traditional knowledge graphs, which is a major strength of KGGen. As seen in Figures 6b and 6e, GraphRAG often extracts very few nodes and connections for an entire article. This sparsity results in the omission of critical relationships and information. For compression, Figure 6a and 6d illustrate sections of the KGs generated by KGGen for the same articles. Figure 6c illustrates one of many issues in OpenIE's KGs. Firstly, most node types are hyperspecific, incoherent phrases. Many of these nodes are redundant near-copies of each other, adding unnecessary complexity to the graph. Additionally, as seen in 6f OpenIE primarily uses pattern matching to identify entities, and frequently produces generic nodes such as "it" and "are". Due to their frequency, these nodes, which contain no useful information, often end up as some of the most well-connected nodes in the graph. Consequently, unrelated concepts end up being just two hops apart, linked by paths through nodes like "it" or "are". By contrast, KGGen consistently generates KGs that are informative and coherent, effectively capturing critical relationships and information from the articles.



(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 6: 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.

6.5 A Note on Scaling

A major motivation for the creation of KGGen was to produce graphs where edge types are generalizable, and used more than once when the corpus grows large. To test the re-usability of our relations, we generate three knowledge graphs from text of different sizes: 10 000 characters, 100 000 characters, and 1 000 000 characters and plot the number of edges divided by the number of unique relations. As one can see from Figure 7, KGGen reuses each relation-type an average of 10 times, and the average number of occurrences of each relation-type increases with the size of the corpus. By contrast, GraphRAG reuses each relation type an average of 2 times regardless of the size of the graph. This suggests that the relations in GraphRAG do not generalize as the corpus grows.

6.6 Efficiency and Cost Analysis

To evaluate the practical applicability of KGGen, we analyzed its computational efficiency and cost on a large-scale extraction task. We extracted a KG from the novel *The Name of the Wind* by Patrick Rothfuss, processing text corpora of increasing sizes. Table 2 demonstrates KGGen's de-duplication effectiveness across different scales.

Table 2: KGGen scaling characteristics showing entity and relation de-duplication

Corpus Size (chars)	Pre-Entities	Post-Entities	Entity De-dup Ratio	Pre-Relations	Post-Relations	Relation De-dup Ratio	Pre-Edges	Post-Edges	Edge De-dup Ratio
100	1	1	1.000	1	1	1.000	1	1	1.000
1,000	20	18	0.900	12	12	1.000	9	9	1.000
10,000	90	78	0.867	78	75	0.962	62	57	0.919
100,000	727	604	0.831	926	924	0.998	498	424	0.851
1,000,000	4,602	3,573	0.776	8,137	8,094	0.995	3,180	2,448	0.770

The de-duplication ratios improve with scale, demonstrating the effectiveness of our clustering algorithm. For the complete novel (1M characters), KGGen achieved a 22.4% reduction in entities and 23% reduction in edges through intelligent consolidation. This analysis highlights the contribution of each module: the extraction phase (Steps 1-2) captures comprehensive information from text, while the resolution phase (Step 3) significantly reduces redundancy without information loss. Table 3 presents the computational cost breakdown for processing the entire novel:

Table 3: KGGen cost and throughput analysis for 1M character corpus

Step	Prompt Tokens	Completion Tokens	Total Tokens	Time (s)	Throughput (tokens/s)	Cost ($)
KG Extraction (Steps 1–2)	1.59M	0.63M	2.22M	273	8,139	0.46
Entity/Edge Resolution (Step 3)	2.93M	0.22M	3.15M	279	11,304	0.38
Total	4.52M	0.85M	5.37M	551	9,739	0.84

For comparison, we evaluated GraphRAG on the same corpus: results can be found in Table 4.

Table 4: GraphRAG scaling characteristics on the same corpus

Corpus Size (chars)	Entities	Relations	Edge Types	Time (s)
100	2	1	1	1.89
1,000	4	3	3	3.01
10,000	16	20	20	29.71
100,000	80	100	99	205.12
1,000,000	514	981	966	2,079.17

While GraphRAG is faster on short corpora, its execution time scales superlinearly. The complete extraction time comparison reveals a significant difference: GraphRAG requires 2,319 seconds for the extraction phase alone on the 1M character corpus, compared to KGGen's total processing time of 551 seconds (including both extraction and resolution). Additionally, GraphRAG produces nearly as many relation types (966) as edges (981), indicating minimal relation reuse and poor generalization compared to KGGen's efficient consolidation.

7 Broader Impact and Community Adoption

Our work produces a KG-from-plain-text extractor that helps to solve the KG-scarcity problem. Improved knowledge-graph extractors could lead to the prevalence of structured text, which can help improve factuality and reliability of information retrieval systems. Our open-source implementation has already enjoyed widespread community adoption. **The package has received over 700 Github stars and has been downloaded over 12,000 times since its release.**

8 Limitations and Future Work

Although KGGen holds many advantages over past extraction methods, its graphs still exhibit problems, like over or under de-duplication of entities and relations. Further research into entity resolution could improve the quality of our KGs. Additionally, our benchmarks measure corpora of up to 5M tokens, which does not reflect the size of web-scale text that would be necessary to produce a KG foundation model. Future expansions of our benchmark could focus on larger corpora to better measure the practicality of different extraction techniques.

Domain-specific knowledge extraction presents additional challenges. Fields like medicine and finance require specialized domain knowledge that general-purpose LLMs may lack, potentially limiting extraction quality compared to human experts. While MINE-2 demonstrates KGGen's capability across diverse domains, incorporating domain-specific ontologies could improve extraction precision. Future work could explore adaptive ontology integration to balance structure with completeness.

Acknowledgments and Disclosure of Funding

We acknowledge support from NSF grant numbers DGE-1656518, NSF 2046795 and 2205329, the MacArthur Foundation, Stanford HAI, and Google Incorporated.

References

Gabor Angeli, Arun Tejasvi Chaganty, Angel X. Chang, Kevin Scott Reschke, Julie Tibshirani, Jean Wu, Osbert Bastani, Keith Siilats, and Christopher D. Manning. Stanford's 2013 kbp system. *Theory and Applications of Categories*, 2013. URL https://api.semanticscholar.org/CorpusID:14273633.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034.

Vahan Arsenyan, Spartak Bughdaryan, Fadi Shaya, Kent Small, and Davit Shahnazaryan. Large language models for biomedical knowledge graph construction: Information extraction from emr notes. In *Workshop on Biomedical Natural Language Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:256390090.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

Wikidata contributors. Wikidata: A free collaborative knowledge base, 2024. URL https://www.wikidata.org.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. A study on term weighting for text categorization: A novel supervised variant of tf.idf. In *Proceedings of the 4th International Conference on Data Management Technologies and Applications (DATA 2015)*, 07 2015. doi: 10.5220/0005511900260037.

Junyang Gao, Xian Li, Yifan Ethan Xu, Bunyamin Sisman, Xin Luna Dong, and Jun Yang. Efficient knowledge graph accuracy evaluation. *ACM Transactions on Information Systems*, 36(2):1–21, 2018. doi: 10.14778/3342263.3342642. URL https://dl.acm.org/doi/pdf/10.14778/3342263.3342642. Duke University and Amazon.com.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, pages 639–648. ACM, 2020. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

Nicolas Heist, Sven Hertling, and Heiko Paulheim. Kgreat: A framework to evaluate knowledge graphs via downstream tasks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*, pages 3938–3942. ACM, 2023. doi: 10.1145/3583780.3615241. URL https://doi.org/10.1145/3583780.3615241. Published on 21 October 2023.

Subhi Issa, Onaopepo Adekunle, Fayçal Hamdi, Samira Si-Said Cherfi, Michel Dumontier, and Amrapali Zaveri. Knowledge graph completeness: A systematic literature review. *IEEE Access*, 9: 31322–31339, 2021. doi: 10.1109/ACCESS.2021.3056622. URL https://ieeexplore.ieee.org/document/9344615.

Shengbin Jia, Yang Xiang, Xiaojun Chen, Kun Wang, and Shijia. Triple trustworthiness measurement for knowledge graph. In *Proceedings of the World Wide Web Conference (WWW '19)*, pages 2865–2871. ACM, May 2019. doi: 10.1145/3308558.3313586. URL https://doi.org/10.1145/3308558.3313586.

Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. From human experts to machines: An llm supported approach to ontology and knowledge graph construction. *ArXiv*, abs/2403.08345, 2024. URL https://api.semanticscholar.org/CorpusID:268379482.

Jonathan Larson and Steven Truitt. Graphrag: Unlocking llm discovery on narrative private data. https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/, 2024. Microsoft Research Blog, published Feb 13, 2024.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015. doi: 10.3233/SW-140.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015. ISBN 0262511290.

Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16:72–79, 03 2001. doi: 10.1109/5254.920602.

Idza Aisara Norabid and Fariza Fauzi. Rule-based text extraction for multimodal knowledge graph. *International Journal of Advanced Computer Science and Applications*, 2022. URL https://api.semanticscholar.org/CorpusID:249304784.

Prakhar Ojha and Partha Talukdar. KGEval: Accuracy estimation of automatically constructed knowledge graphs. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1750, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1183. URL https://aclanthology.org/D17-1183/.

Sergio Oramas, Mohamed Sordo, and Luis Espinosa-Anke. A rule-based approach to extracting relations from music tidbits. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 661–666, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334730. doi: 10.1145/2740908.2741709. URL https://doi.org/10.1145/2740908.2741709.

Yifan Qi, Weiguo Zheng, Liang Hong, and Lei Zou. Evaluating knowledge graph accuracy powered by optimized human-machine collaboration. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, pages 1368–1378. ACM, 2022. doi: 10.1145/3534678.3539233. URL https://doi.org/10.1145/3534678.3539233.

Bin Qiao, Zhiliang Zou, Yurong Huang, Buyue Wang, and Changlong Yu. A joint model for entity and relation extraction based on BERT. *Neural Computing and Applications*, 34(5):3471–3483, 2022. ISSN 1433-3058. doi: 10.1007/s00521-021-05815-z. URL https://doi.org/10.1007/s00521-021-05815-z.

Edward W. Schneider. Course modularization applied: The interface system and its implications for sequence control and data analysis. In *Association for the Development of Instructional Systems (ADIS)*, Chicago, Illinois, April 1973. Presented in April 1972.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the ACL and the 12th International Joint Conference on Natural Language Processing (AACL-IJCNLP 2022)*, 11 2022. doi: 10.18653/v1/2022.aacl-main.46.

Kartik Shenoy, Filip Ilievski, Daniel Garijo, Daniel Schwabe, and Pedro Szekely. A study of the quality of wikidata. *Journal of Web Semantics, Special Issue on Community-Based Knowledge Bases*, 06 2021. doi: 10.48550/arXiv.2107.00156.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936547. doi: 10.1145/1242572.1242667. URL https://doi.org/10.1145/1242572.1242667.

Jiajun Tan, Dong Wang, Jingyu Sun, Zixi Liu, Xiaoruo Li, and Yang Feng. Towards assessing the quality of knowledge graphs via differential testing. *Available online, Version of Record*, 2024. URL `https://doi.org/10.1016/j.jss.2024.07.005`. Received 3 October 2023, Revised 15 June 2024, Accepted 26 June 2024, Available online 29 June 2024.

Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5(11):1483–1494, July 2012. ISSN 2150-8097. doi: 10.14778/2350229.2350263. URL `https://doi.org/10.14778/2350229.2350263`.

Bingcong Xue and Lei Zou. Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4969–4988, May 2023. ISSN 1041-4347. doi: 10.1109/TKDE.2022.3150080. URL `https://doi.org/10.1109/TKDE.2022.3150080`. Published on 10 February 2022.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1237. URL `https://aclanthology.org/D15-1237/`.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL `https://api.semanticscholar.org/CorpusID:268987666`.

Ruiqi Zhu, Alan Bundy, Jeff Pan, Kwabena Nuamah, Fangrong Wang, Xue Li, Lei Xu, and Stefano Mauceri. Assessing the quality of a knowledge graph via link prediction tasks. In *Proceedings of the 7th International Conference on Natural Language Processing and Information Retrieval (NLPIR 2023)*, pages 1–10, Seoul, Republic of Korea, December 2023. ACM. doi: 10.1145/3639233.3639357. URL `https://doi.org/10.1145/3639233.3639357`. School of Informatics, University of Edinburgh, United Kingdom; Huawei Ireland Research Centre, Ireland.

A Prompts for KG Extraction

This section provides the exact prompts used to extract KG's from the text.

The initial KG was extracted using the following two prompts passed as DSPy signature descriptions.

Prompt for extracting entities: Extract key entities from the source text. Extracted entities are subjects or objects. This is for an extraction task, please be thorough and accurate to the reference text.

Prompt for extracting relations: Extract subject-predicate-object triples from the source text. Subject and object must be from entities list. Entities provided were previously extracted from the same source text. This is for an extraction task, please be thorough, accurate, and faithful to the reference text.

After extracting the entities and relations from each unit of text, we begin the de-duplication process, which is performed using the following prompts.

Prompt for entity or edge resolution:
Find duplicate {item_type} for the item and an alias that best represents the duplicates. Duplicates are those that are the same in meaning, such as with variation in tense, plural form, stem form, case, abbreviation, shorthand. Return an empty list if there are none.

B Prompts for MINE

This section provides the LLM prompts used by MINE to evaluate KGs.

Prompt for extracting a fact from article: Extract 15 basic, single pieces of information from the following text that describe how one object relates to another. Present the pieces of info in short sentences and DO NOT include info not directly present in the text. Your output should be of the form ["info1", "info2" ,..., "info15"]. "Make sure the strings are valid Python strings."

Prompt for evaluating if a fact is contained in the query result:
ROLE: "You are an evaluator that checks if the correct answer can be deduced from the information in the context.
TASK: Determine whether the context contains the information stated in the correct answer.
Respond with "1" if yes, and "0" if no. Do not provide any explanation, just the number.

Prompt for RAG response:

```
Use the following knowledge graph triples and text evidence to
answer the question.
Triples:  {triples_text}
Text Evidence:  {text_block}
Question:  {query} Answer:
```

Prompt for Evaluating containment of WikiQA answer:

```
You are a fact-checking assistant
Question:  question
Expected answer:  expected
Model’s response:  response
Does the model’s response contain the information in the expected
answer?
Respond with one word:  Yes or No.
```

C Example Article from MINE

This section provides the article that the example fact is from.

Title: The Rise of Cryptocurrencies

Content: Cryptocurrencies have taken the financial world by storm in recent years, revolutionizing the way we think about money and transactions. From the creation of Bitcoin in 2009 by an anonymous individual or group known as Satoshi Nakamoto, to the thousands of altcoins that have since emerged, cryptocurrencies have become a significant player in the global economy.One of the key factors contributing to the rise of cryptocurrencies is the decentralized nature of these digital assets. Unlike traditional fiat currencies that are controlled by governments and central banks, cryptocurrencies operate on a peer-to-peer network, allowing for transactions to occur directly between users without the need for intermediaries. This decentralization not only provides users with more control over their funds but also enhances security and privacy.Another driving force behind the popularity of cryptocurrencies is the technology that underpins them - blockchain. Blockchain is a distributed ledger technology that ensures the transparency and immutability of transactions on the network. Each transaction is recorded in a block and linked to the previous block, forming a chain of blocks that cannot be altered once validated by the network. This technology has been instrumental in building trust and confidence in cryptocurrencies, as it eliminates the need for a trusted third party to oversee transactions. The concept of decentralization and blockchain technology has also paved the way for various applications beyond just digital currencies. Smart contracts, for example, are self-executing contracts with the terms of the agreement directly written into code. These contracts automatically enforce and execute themselves when predefined conditions are met, eliminating the need for intermediaries and streamlining processes in various industries. Cryptocurrencies have also gained traction due to their potential for financial inclusion. In many parts of the world, traditional banking services are inaccessible or too costly for a significant portion of the population. Cryptocurrencies offer a way for individuals to access financial services, such as transferring money and making payments, without the need for a traditional bank account. This has the potential to empower individuals in underserved communities and drive economic growth. The volatile nature of cryptocurrencies has attracted both investors seeking high returns and speculators looking to capitalize on price fluctuations. The rapid appreciation of certain cryptocurrencies, such as Bitcoin, has led to a surge in interest from retail and institutional investors alike. While this volatility presents opportunities for profit, it also poses risks, as prices can fluctuate dramatically in a short period. Regulation has been a contentious issue in the cryptocurrency space, with governments and regulatory bodies grappling with how to oversee this emerging asset class.

```
Some countries have embraced cryptocurrencies and blockchain
technology, recognizing their potential for innovation and
economic growth.  Others have taken a more cautious approach,
citing concerns about money laundering, tax evasion, and consumer
protection.  Despite the challenges and uncertainties surrounding
cryptocurrencies, their rise has been undeniable.  As more
individuals and businesses adopt digital currencies for transactions
and investments, the landscape of finance is evolving rapidly.  The
future of cryptocurrencies remains uncertain, but their impact on
the financial world is already profound.  In conclusion, the rise
of cryptocurrencies can be attributed to their decentralized nature,
blockchain technology, financial inclusion potential, investment
opportunities, and regulatory challenges.  As these digital assets
continue to gain acceptance and adoption, they are reshaping the
way we think about money and finance.  Whether cryptocurrencies will
become mainstream or remain on the fringes of the financial system
remains to be seen, but their impact is undeniable and will likely
continue to unfold in the years to come.
```

D Additional Graphs

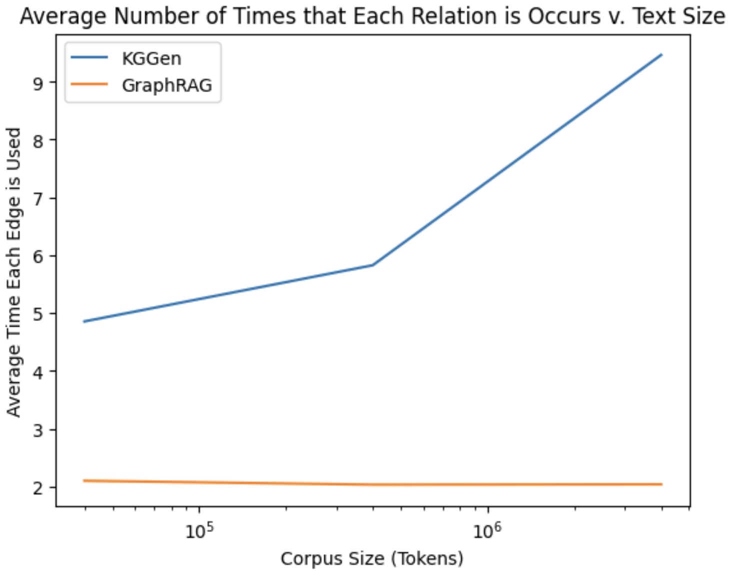


Figure 7: As graphs increase in size, KGGen tends to reuse each unique relation type more frequently, while GraphRAG maintains a consistent average usage of about 2 instances per relation type regardless of graph size.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We support all claims made in the abstract: we build a knowledge-graph extractor and benchmark its performance against existing knowledge-graph extractors.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a limitations section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not provide theoretical claims in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly describe all methodology and will release code with the final version along with a python module to perform KG extraction.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide code in the supplementary material, and release all code publicly.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all information necessary to replicate the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report statistical error bars where possible for our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiments do not require special hardware, and can be run on most laptops with production models. They require only an API key from a model provider– this is clear from the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is very ethical, and does not breach any of the NeurIPS ethics guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a broader impacts section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release new models or sensitive data. Our code uses mostly existing data, and we release a small number of new data that we have checked is harmless.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit everyone whose code or ideas we use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our assets include meticulous documentation for our new package and dataset, which will be released with the final paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not crowd source.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: We do not have human test subjects and do not require IRB approval.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the role of LLMs in our methodology very clearly.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.