Exploring LLM To Extract Knowledge Graph From Academic Abstracts

Victor Eiti Yamamoto[1,2,*], Othmane Kabal[3], Lakshan Karunathilake[1,2], Kotaro Nishigori[1,2], Vicente Lermanda[4], Shixiong Zhao[1,2], Hiroki Uematsu[1,2], Yanming He[1,2] and Hideaki Takeda[1,2]

[1]National Insitute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
[2]The Graduate University for Advanced Studies, SOKENDAI, Shonan Village, Hayama, Kanagawa 240-0193 Japan
[3]Nantes University, LS2N, Nantes 44300, France
[4]Pontifical Catholic University of Chile, Avenida Libertador General Bernardo O'Higgins #340, Santiago, Santiago Metropolitan Region, Chile

Abstract

Knowledge graphs (KGs) are a powerful tool for representing semantic information. Existing methods depend on the use of human annotation or semi-structured automated methods based on basic metadata. However, academic papers and their abstracts are still the main way to carry academic information. The development of LLM leads to new tools to solve semantically heavy problems, so LLM can help to create KGs from texts automatically. This study comparatively evaluated LLMGraphTransformer, KGGen, and GT2KG, three LLM-based KG construction methods, using three computer science abstracts. We assessed performance via precision, recall, and F1-score against a gold standard and analyzed differences in knowledge representation. Our findings revealed a trade-off between precision and recall in the extracted triples. Furthermore, GT2KG extracted hierarchical and definitional triples, whereas LLMGraphTransformer and KGGen Pro identified causal and functional relationships. Divergent predicate structures—simple in the gold standard vs. complex in some LLM outputs—suggest varied KG objectives, from traditional knowledge sharing to Retrieval-Augmented Generation (RAG) context capture. These results indicate that LLM-based KG construction is promising but requires further research to enhance accuracy and robustness, emphasizing that methodology choice should align with the intended application.

Keywords

Knowledge Graph extraction, Knowledge Graph construction, Large language model

1. Introduction

Knowledge graphs (KGs) are essential for structuring information to support complex problem-solving in intelligent systems [1]. Within the academic domain, however, existing KGs are often constructed from basic metadata like authors and institutions, neglecting the rich scientific discourse—including methods, findings, and hypotheses—embedded within the full text of articles. This limitation hinders the development of systems that can truly comprehend and reason about scientific contributions.

The recent advancements in Large Language Models (LLMs) offer a powerful solution for extracting granular entities and relationships directly from unstructured text. This paper presents a comparative evaluation of state-of-the-art LLM-based methods for automatically constructing KGs from scientific papers, bridging the gap between metadata-level graphs and deep content understanding.

We assessed three distinct KG construction methods: LLMGraphTransformer [2], KGGen [3], and GT2KG [4]. To evaluate their performance, we adopted the framework proposed by Kebal et al. [4]. Following this, we manually aligned the triples generated by each method against a gold standard

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
eitiyamamoto@nii.ac.jp (V. E. Yamamoto); othmane.kabal@univ-nantes.fr (O. Kabal); lakshan@nii.ac.jp (L. Karunathilake); watagori@nii.ac.jp (K. Nishigori); vlermanda@uc.cl (V. Lermanda); shixiong@nii.ac.jp (S. Zhao); uematsu@alchembright.com (H. Uematsu); yanming@nii.ac.jp (Y. He); takeda@nii.ac.jp (H. Takeda)
https://github.com/eitiyamamoto (V. E. Yamamoto)
0000-0002-3825-6461 (V. E. Yamamoto); 0000-0002-6120-1450 (K. Nishigori); 0000-0002-3158-3873 (S. Zhao); 0000-0003-4215-3112 (H. Uematsu); 0000-0002-2909-7163 (H. Takeda)

dataset and computed their precision, recall, and F1-score. Furthermore, we conducted a comparative analysis of the generated triples. We specifically analyzed the resulting KG to elucidate the fundamental differences in how each tool defines and structures knowledge, revealing their distinct representational approaches.

Our evaluation reveals a clear trade-off between precision and recall across the tested methods. While GT2KG achieved the highest precision, LLMGraphTransformer demonstrated superior recall. Ultimately, LLMGraphTransformer (with Gemini Flash) and KGGen (with Gemini Pro) obtained the highest F1-scores, representing the most effective balance of precision and recall.

The comparable performance across all methods indicates that while LLM-based KG construction is promising, significant challenges remain. This highlights a critical need for further research to improve the accuracy and robustness of automated knowledge extraction from complex scholarly texts.

2. Related Works

Various approaches have been proposed to convert natural language into knowledge graphs. Early work by Hearst, developed in the early 1990s, introduced rule-based and mechanical transformation methods for information extraction [5]. This line of research focused on deterministic patterns to identify relations in natural language, providing the foundation for later approaches. In the late 2000s, Mintz et al. expanded on this direction by applying syntactic parsing combined with machine learning techniques, such as distant supervision, to automatically generate training data for relation extraction tasks [6]. This significantly reduced the need for manual annotation and enabled scalable learning. By the mid-2010s, Zeng et al. introduced deep learning-based models for relation classification. Their use of convolutional neural networks (CNNs) demonstrated that automatically learned features could outperform traditional hand-crafted ones, marking a shift towards end-to-end neural methods [7].

Pretrained language models (PLMs) have been employed for knowledge graph construction, as demonstrated in works such as [8] and [9]. In addition, generative models have been utilized for knowledge graph completion (KGC), as explored in [10], [11], and [12]. GraphRAG[13] constructs knowledge graphs from unstructured text to curate and summarize information, enabling more accurate and semantically grounded retrieval.

Alongside construction, a parallel line of work asks how to evaluate the resulting KGs. Benchmark-based evaluations rely on curated datasets: CARB provides a crowdsourced OpenIE benchmark for triple extraction quality [14]; WebNLG measures graph–text fidelity via RDF-to-text generation [15]; DocRED [16] and TACRED [17] target the relation extraction task.

Beyond static benchmarks, other evaluation approaches were proposed. Differential testing [18] trains multiple KG embedding models, runs head-prediction, and computes a differential score from the proximity of model outputs. Another, downstream-utility evaluation [19], judges a KG by how much it improves fixed tasks (e.g., classification, clustering, recommendation). Finally, an LLM-as-judge approach has been proposed, where GraphJudge [20] first filters noise with an entity-centric strategy and then uses a fine-tuned LLM to assess the correctness and consistency of triples and entities.

3. Approach

We evaluated several LLM-based tools for knowledge graph construction from text using three randomly selected abstracts (Document IDs 23, 438, and 519) from the G-T2KG Computer Science benchmark [4]. This benchmark comprises 12 curated abstracts (108 sentences) selected from diverse topics to ensure a variety of terminology and writing styles. For each abstract, we ran the tools to generate triples (subject, predicate, object) and compared the output against the gold-standard triples from the benchmark.

The evaluation was conducted manually, focusing on the semantic equivalence of the generated triples. Specifically, a predicted triple was considered correct if it semantically matched a gold-standard triple, regardless of differences in wording (e.g., synonyms), structure (e.g., active/passive voice), or morphology (e.g., plural/singular forms).

We select three methods to compare: LLMGraphTransformer, KGGen and G-T2KG. LLMGraphTransformer from LangChain transforms text into a KG by employing a pipeline of predefined prompts to extract entities as nodes and their corresponding relationships as edges [2]. These tools were selected based on three criteria: (1) their foundation in LLMs; (2) the replicability of their tests in our environment; and (3) the structural alignment of the generated KG with the gold standard. Specifically, both employ a triple structure where each element consists of a few words, not a phrase. KGGen also leverages an LLM for KG extraction but introduces a clustering step to produce a denser graph [3]. Its process involves predicting triples with an LLM-based extractor, clustering these triples for refinement, and performing entity resolution to merge nodes that refer to the same concept (e.g., handling plurals and capitalization). Similarly, G-T2KG combines the OpenIE framework [21] with noun phrase-based cleaning and LLM-based validation to reduce irrelevant triples and mitigate LLM hallucinations [4].

Our experimental setup involved the following models: Gemini 2.5 Flash was used for LLMGraphTransformer, while KGgen was tested with both Gemini 2.5 Flash and Gemini 2.5 Pro. For a baseline comparison, the results for the G-T2KG method, which employed the GPT-4 model, were extracted from its source publication.

4. Result

Table 1 presents the performance of each model across the three tested abstracts. GT2KG achieved the highest precision on two datasets, whereas LLMGraphTransformer was top on one. For both recall and F-measure, LLMGraphTransformer and KG-Gen Pro outperformed the other models on two datasets, while GT2KG also secured the highest F-measure on a single dataset.

Table 1
Model Performance Across Different Datasets

Abstract	Model	Precision	Recall	F-measure
26	LLMGraphTransformer	**0.60**	**0.55**	**0.57**
	KG-Gen Flash	0.15	0.18	0.17
	KG-Gen Pro	0.30	**0.55**	0.39
	GT2KG	0.14	0.09	0.11
438	LLMGraphTransformer	0.47	**0.53**	**0.50**
	KG-Gen Flash	0.40	0.40	0.40
	KG-Gen Pro	0.54	0.47	**0.50**
	GT2KG	**0.57**	0.27	0.36
519	LLMGraphTransformer	0.48	0.43	0.45
	KG-Gen Flash	0.37	0.48	0.42
	KG-Gen Pro	0.41	**0.74**	**0.53**
	GT2KG	**0.82**	0.39	**0.53**

Table 2, which shows the gold standard triples for abstract 438, reveals distinct performance patterns among the evaluated methods. LLMGraphTrans and KGGen Pro demonstrate similar capabilities, successfully identifying a comparable set of triples related to causal and functional relationships.

In contrast, G-T2KG provides complementary results, uniquely finding hierarchical and definitional triples (e.g., using the is-a predicate) that all other methods missed. This highlights a clear difference in the types of knowledge each system can extract.

Furthermore, consensus across all methods was minimal. Only a single triple (Thyristor-Controlled Braking Resistor, stabilize, target generator) was identified by every system. Conversely, a shared limitation was also evident, as none of the methods managed to extract the three triples indicating the causes of a critical issue.

Table 2
Comparison of Knowledge Graph Generation Methods for abstract 438

Subject	Predicate	Object	LGT	KGF	KGP	G2K
conventional power system	turn-to	smart grid	✓		✓	
SG cyber security	is-a	critical issue				✓
interconnection of several load	causes	critical issue				
generator	causes	critical issue				
renewable resource	causes	critical issue				
cyber-physical attack	skos:broader	threatening of SGs security	✓	✓	✓	
cyber-physical attack	causes	blackout	✓	✓		
cyber-physical attack	causes	destruction of infrastructure	✓	✓		
cyber-physical attack	includes	cyber switching attack	✓		✓	
cyber-physical attack	have	severity				✓
cyber switching attack	destabilise	smart grid	✓		✓	
Thyristor-Controlled Braking Resistor	mitigate	attack	✓	✓	✓	
Thyristor-Controlled Braking Resistor	stabilize	target generator	✓	✓	✓	✓
large blackout	is-a	severe consequence				✓
destruction of infrastructures	is-a	severe consequence				✓

Note: LGT: LLMGraphTrans; KGF: KGGen Flash; KGP: KGGen Pro; G2K: G-T2KG. The ✓ symbol indicates presence.

5. Discussion

As shown in Table 1, LLMGraphTransformer and KG-Gen Pro demonstrate the best balance between precision and recall, achieving the highest recall in most cases while maintaining comparable precision. Conversely, G-T2KG exhibits the highest overall precision but suffers from low recall in most instances.

A comparison between KGGen Flash and KGGen Pro suggests that larger LLM models can lead to improved performance. However, LLMGraphTransformer, which also utilizes Gemini Flash, achieved similar results, indicating that the choice of methodology is as crucial as the model size.

G-T2KG, relying on an initial extraction phase with OpenIE, offers the key advantage of strictly adhering to information explicitly stated in the text, thereby effectively preventing hallucinations. However, it struggles with the accurate identification of entities and predicates, often leading to truncated or overly extended entity spans. This limitation persists even after cleaning algorithms are applied, particularly in complex sentences involving conjunctions or coreference. In contrast, LLMs, leveraging their deep semantic understanding, are generally more effective at capturing relevant entities in diverse and intricate contexts. However, this capability can come at the cost of lower precision and potential hallucinations.

A manual comparison of the generated triples against the gold standard is necessary, as the terminology adopted by each system can vary significantly. For instance, the gold standard almost exclusively employs predicates consisting of a single verb. In contrast, systems such as KGGen and G-T2KG generate more complex, compound-verb predicates. These are challenging to normalize and unify for downstream applications, including knowledge graph-based search systems. This divergence in predicate structure likely originates from the fundamental purpose for which KGs are created. As discussed by Hogan et al. [13], KGs were traditionally created to facilitate knowledge sharing within specific organizations or communities. Conversely, recent LLM-based approaches often generate KGs with the primary goal of capturing contextual information for applications like Retrieval-Augmented Generation (RAG) [3]. Therefore, the intended application should guide the selection of a KG creation methodology, taking this fundamental difference in purpose into account.

6. Conclusion

In this research, we investigated three different models that leverage LLMs to generate KGs from text. We evaluated their performance using abstracts from scientific papers to determine their capability to capture rich semantic context. Our findings indicate that these models show considerable promise,

though there are clear areas for improvement. Furthermore, we discovered that different triple extraction methods generate distinct yet complementary sets of triples. This suggests that LLM-based approaches can serve a different purpose in KG creation compared to conventional methods. For future work, we plan to extend our evaluation by incorporating a wider range of methods and LLMs, testing on abstracts from diverse scientific domains, and extending the analysis to multiple languages.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini and Grammarly in order to: paraphrase and reword, improve writing style, and grammar and spelling checking. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] X. Du, N. Li, Academic paper knowledge graph, the construction and application., in: ICBASE, 2022, pp. 15–27.

[2] GitHub - langchain-ai/langchain: Build context-aware reasoning applications — github.com, https://github.com/langchain-ai/langchain, 2025. [Accessed 28-07-2025].

[3] B. Mo, K. Yu, J. Kazdan, P. Mpala, L. Yu, C. Cundy, C. Kanatsoulis, S. Koyejo, Kggen: Extracting knowledge graphs from plain text with language models, arXiv preprint arXiv:2502.09956 (2025).

[4] O. Kabal, M. Harzallah, F. Guillet, R. Ichise, Enhancing domain-independent knowledge graph construction through openie cleaning and llms validation, Procedia Computer Science 246 (2024) 2617–2626.

[5] M. A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics, 1992. URL: https://aclanthology.org/C92-2082/.

[6] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: K.-Y. Su, J. Su, J. Wiebe, H. Li (Eds.), Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Association for Computational Linguistics, Suntec, Singapore, 2009, pp. 1003–1011. URL: https://aclanthology.org/P09-1113/.

[7] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: J. Tsujii, J. Hajic (Eds.), Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 2335–2344. URL: https://aclanthology.org/C14-1220/.

[8] S. Hao, B. Tan, K. Tang, H. Zhang, E. P. Xing, Z. Hu, Bertnet: Harvesting knowledge graphs from pretrained language models, arXiv preprint arXiv:2206.14268 (2022).

[9] V. Swamy, A. Romanou, M. Jaggi, Interpreting language models through knowledge graph extraction, arXiv preprint arXiv:2111.08546 (2021).

[10] B. R. Andrus, Y. Nasiri, S. Cui, B. Cullen, N. Fulda, Enhanced story comprehension for large language models through dynamic document-based knowledge graphs, in: Proceedings of the AAAI conference on artificial intelligence, volume 36, 2022, pp. 10436–10444.

[11] H. Chen, X. Shen, Q. Lv, J. Wang, X. Ni, J. Ye, Sac-kg: Exploiting large language models as skilled automatic constructors for domain knowledge graphs, arXiv preprint arXiv:2410.02811 (2024).

[12] Y. Lairgi, L. Moncla, R. Cazabet, K. Benabdeslem, P. Cléau, itext2kg: Incremental knowledge graphs construction using large language models, in: International Conference on Web Information Systems Engineering, Springer, 2024, pp. 214–229.

[13] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (Csur) 54 (2021) 1–37.

[14] S. Bhardwaj, S. Aggarwal, et al., Carb: A crowdsourced benchmark for open ie, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6262–6267.

[15] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, The webnlg challenge: Generating text from rdf data, in: Proceedings of the 10th International Conference on Natural Language Generation, ACL Anthology, 2017, pp. 124–133.

[16] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, M. Sun, Docred: A large-scale document-level relation extraction dataset, arXiv preprint arXiv:1906.06127 (2019).

[17] Y. Zhang, V. Zhong, D. Chen, G. Angeli, C. D. Manning, Position-aware attention and supervised data improve slot filling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

[18] J. Tan, D. Wang, J. Sun, Z. Liu, X. Li, Y. Feng, Towards assessing the quality of knowledge graphs via differential testing, Information and Software Technology 174 (2024) 107521.

[19] N. Heist, S. Hertling, H. Paulheim, Kgreat: A framework to evaluate knowledge graphs via downstream tasks, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 3938–3942.

[20] H. Huang, C. Chen, C. He, Y. Li, J. Jiang, W. Zhang, Can llms be good graph judger for knowledge graph construction?, arXiv preprint arXiv:2411.17388 (2024).

[21] J. L. Martinez-Rodriguez, I. López-Arévalo, A. B. Rios-Alvarado, Openie-based approach for knowledge graph construction from text, Expert Systems with Applications 113 (2018) 339–355.