Enhancing Domain-Independent Knowledge Graph Construction through OpenIE Cleaning and LLMs Validation

Othmane Kabal, Mounira Harzallah, Fabrice Guillet, Ryutaro Ichise

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

Enhancing Domain-Independent Knowledge Graph Construction through OpenIE Cleaning and LLMs Validation

Othmane KABAL[a,*], Mounira HARZALLAH[a], Fabrice GUILLET[a], Ryutaro ICHISE[b]

[a]*Nantes University, LS2N, Nantes 44300, France*
[b]*Tokyo Institute of Technology, Tokyo, Japan*

Abstract

In the challenging context of Knowledge Graph (KG) construction from text, traditional approaches often rely on Open Information Extraction (OpenIE) pipelines. However, they are prone to generating many incorrect triplets. While domain specific Named Entity Recognition (NER) is commonly used to enhance the results, it compromises the domain independence and misses crucial triplets. To address these limitations, we introduce *G-T2KG*, a novel pipeline for KG construction that aims to preserve the domain independence while reducing incorrect triplets, thus offering a cost-effective solution without the need for domain-specific adaptations. Our pipeline utilizes state-of-the-art OpenIE combined with both a noun phrase-based cleaning and a LLMs based validation. It is evaluated using gold standards in two distinct domains (i.e., computer science and music) that we have constructed in the context of this study. On computer science corpus, the experimental results demonstrate a higher recall as compared to state-of-the-art approaches, and a higher precision notably increased by the integration of LLMs. Experiments on the music corpus show good performance, underscoring the versatility and effectiveness of *G-T2KG* in domain-independent KG construction.

Keywords: Knowledge graph; From raw text ; Domain-independent building; Knowledge graph construction pipeline; LLMs based validation; Information Extraction; LLMs for KG.

1. Introduction

The Semantic Web's core objective is to facilitate the extraction and formalization of information into structured formats, among which knowledge graphs (KGs) stand out as a cornerstone in various modern applications, including question answering systems [25] and recommendation engines [22]. Stimulated by the exponential growth of textual data, this paper adresses the challenging issue of KG construction from raw texts.
KGs organize information into triplets of (subject, predicate, object), linking entities through semantic relationships

* Corresponding author. Tel.: +33745679512
E-mail address: othmane.kabal@univ-nantes.fr

[10]. KG construction pipeline is composed of several stages : 1) Named Entity Recognition (NER), 2) Relation Extraction (RE), 3) Entity and Predicate Mapping. Manual construction, while precise, is prohibitively expensive and requires substantial expertise [12].
Recent improvements in NLP tasks make automatic KG construction an interesting alternative; however, it is limited both by the complexity of natural language as well as the specificity of domain knowledge.

Firstly, in order to deal with language complexity, some approaches focus on domain-specific KGs [1], by specializing KG construction steps such as NER and RE among identified entities. However, these approaches often miss many triplets and suffer from limited reusability due to the specialized adaptation they require, including the need for domain-specific annotated data for effective NER and RE. Secondly, as an alternative, Open Information Extraction (OpenIE) methods have been explored for their potential to extract domain-independent triplets. While these capture a larger part of relevant triplets, they also deliver many irrelevant ones requiring post-filtering typically by retaining only those triplets involving entities recognized by NER. This not only compromises OpenIE's domain independence, but also entails the risk of losing valuable information. In parallel, Large Language Models (LLMs) like GPT-4 [2] and Llama [29], known for their generative capabilities, have been applied to KG construction. Despite their domain-agnostic potential, they face challenges such as hallucination, which limits their reliability.

More precisely, in order to overcome these limitations, we propose a new domain independent approach, so called *General T2KG (G-T2KG)*, that incorporates both a noun phrase based cleaning and a LLMs based validation. The remainder of the paper is structured as follows: Section 2 discusses state-of-the-art approaches for KG construction. In Section 3, we introduce our approach, emphasizing its relevance. Section 4 presents the benchmarks in music and computer science, along with evaluation and comparison results against other approaches (e.g., SciCero approach, and GPT-4). Finally, the paper concludes with some closing remarks and outlines future work in Section 5.
We have made the source code of our approach, as well as the gold standards we developed, available to the scientific community through a GitHub repository[1].

2. Related Work

With the advances in NLP techniques and the introduction of multi-task models[8], approaches to KG generation from texts can be divided into two main categories: 1) *Pipeline* approaches that decompose the KG generation process into multiple sub-components and employ a specific model for each ; and 2) *LLM* based approaches that utilize a single model (e.g., GPT, Llama)[29] for the whole process. Each approach differs from the others in the way it organizes its components, its reliability and whether or not it is domain-dependent.

Domain-independent pipeline approaches. Text2KG [13] is designed for constructing KGs from Wikipedia sentences. It is composed of 3 consecutive stages : a coreference resolution stage [16], a mapping stage with entities of an external knowledge base (KB) (i.e. DBpedia) [19], and then a binary relation extraction stage with an OpenIE system, OLLIE [27]. Another approach also relies on OpenIE for KG generation [18] that emphasize entity linking, utilizing a collective intelligence from three distinct tools: TagMe [11], Spotlight [19], and Babelfy [21] to identify a set of entities, then, these entities are associated with corresponding noun phrases (NPs) to enhance the coherence of the represented facts in RDF format. The strength of domain-independent methods lies in their lack of domain knowledge requirements, alongside their reusability and cost-effective adaptation to specific domains. However, being general in nature and reliant on OpenIE for discovering a wide range of triplets without prior knowledge, these methods often produce a significant number of incorrect triplets. For instance, in [13], if an object within a triplet is not mapped with an entity in DBpedia, it is still retained as a literal. This can lead to incorrect object entities. Moreover, triplets filtering based on KB mapping as in [18], exhibit another weakness: the potential to overlook many relevant triplets which refer to entities or relationships not represented in the KB. These two examples are highlighting a critical challenge in balancing comprehensiveness and accuracy.

Domain-dependent approaches. Doc2KG [28] creates KGs from administrative documents using a specialized approach rather than OpenIE. It employs Semantic Rule Labeling (SRL) to identify verb arguments [23], and identifies

[1] https://github.com/OthmaneKabal/G-T2KG

entities using a domain-specific NER system [4]. It retains only triplets which elements are named entities or proper nouns part of the identified entities.

A recent sophisticated and comprehensive approach called SciCero[9] aims to construct a computer science knowledge graph (KG) from paper abstracts and titles has been conducted in contrast to previously presented methods [13, 18, 28]. This approach employs multiple techniques to detect entities [26, 30] and extract triplets using an OpenIE system [17], the DygIEpp tool [30], and pattern based component,thereby avoiding information loss. A double mapping for entities using external resources (wikidata and dbpedia) and transformer based method are performed aiming to reduce the heterogeneity of entities. For the predicates, only a predefined dictionary of verbs was used. Finally, a validation component verifies the correctness of the generated triplets using a classifier based on Sci-BERT [5] that improve significantly the precision.
Domain-specific approaches yield precise KGs, yet their major limitation lies in reusability and adaptability to other domains modifying each component for different applications proves challenging, resource-intensive, and time-consuming. Additionally, systems such as NER, and RE necessitate annotated data, further compounding the resource demands.
Large Language Models (LLMs). LLMs have demonstrated strong performance across various NLP tasks through the use of relevant prompts including the field of Information Extraction [31]. We can distinguish two approaches in prompting for KG construction: a single prompt to generate the final KG, or a distinct prompt for each sub-task. [20] performs single prompt on Vicuna-13B and Alpaca-LoRA-13B that are fine-tuned based on Llama [29] and employ a few-shot prompting technique. The prompt contains examples to enhance contextual understanding, and an ontology to specify the concepts and relationships to be extracted from a given sentence. The main weakness of this method is the limited scope of the ontologies used, coupled with the application of generative models for complex tasks where hallucination can occur, impacting the quality of the graph by introducing incorrect facts. Moreover, performance issues arise, especially in highly specific domains where LLMs, including GPT-3 and GPT-4, often err. This is evidenced by research [32] employing various prompting techniques like zero-shot and one-shot for KG construction tasks, which revealed that these models underperform in specialized domains (e.g., SciERC dataset) compared to fine-tuned models. On another note, a study has employed GPT-3 in an iterative approach, breaking down the problem into simpler tasks with different prompts based on OpenAI's guidelines [2] which suggest that LLMs excel in simpler tasks. This strategy, used by [7], maintains the connection between prompts by incorporating the context of previous steps in each phase, highlighting a method to mitigate some limitations of LLMs in complex information extraction tasks for KG construction.

In comparison, our pipeline approach General-T2KG addresses the aforementioned challenges through a hybrid, domain-independent approach that performs well across various domains. It leverages the capability of OpenIE to discover a wide range of triplets without prior knowledge, tackling issues such as noisy or lost triplets through NP-based syntactic cleaning of entities. Additionally, it mitigates the problem of incorrect triplets by incorporating a validation component that utilizes the GPT-4 model.

3. Proposed method

General T2KG (G-T2KG) is a pipeline approach that is structured in five components presented in figure 1. *Text preprocessing (C1)* is an initial stage for preparing the corpora by cleaning and resolving coreferences, as well as performing sentence segmentation to facilitate the subsequent component. *Information Extraction (C2)* utilizes two tools for triplets extraction. The first one, Open Information Extraction system(C2.1), is state-of-the-art OpenIE 6 [14], which embeds a deep neural networks providing a balance between speed and accuracy. Additionally, it features an advanced coordination analyzer that efficiently processes conjunctive sentences, extracting multiple triplets when conjunctives are present which improves recall and reduces information loss. The second tool, pattern-based Hypernym Relation Extractor (C2.2), is designed to extract hypernym relationships to enrich our graph with "is-a" relationships that are not explicitly mentioned in the text, using a set of predefined patterns [24]. *Post-processing component (C3)* focuses on cleaning and integrating the extracted triplets through a syntactic rule-based approach (C3.1).

[2] https://platform.openai.com/docs/guides/prompt-engineering

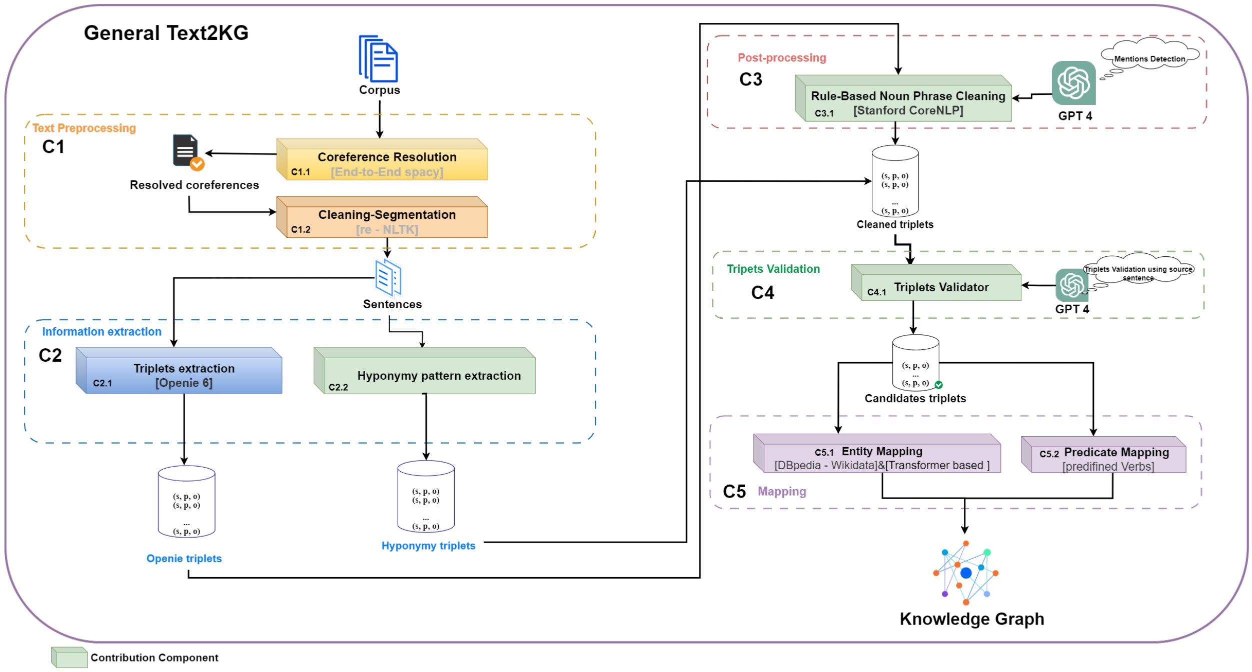

Fig. 1. General T2KG Architecture

It is followed by *triplets validation component (C4)* ensures the correctness of triplets by comparing them with the source sentences using GPT-4. Finally, *the mapping component (C5)*, which includes the mapping of entities (C5.1) and predicates (C5.2) to reduce the heterogeneity (e.i., the same entity/predicate can be referenced by different labels). The mapping approach proposed by SciCero[9] is adopted in the pipeline.
It should be noted that we did not perform the mapping before validation by GPT because if an entity or predicate is mapped to another that does not exist in the sentence, it would complicate the verification task for the model.

In the following, we focus on two sub-components ”Rule-Based Noun Phrase Cleaning” (C3.1) and ”Triplets Validation” (C4.1), which stress the main contributions of this paper, in addition to the proposal of the G-T2KG architecture.

3.1. *Rule-Based Noun phrase Cleaning*

This sub-component C3.1 aims to enhance the quality of the triplets extracted from OpenIE 6 in 3 steps, ensuring they represent real-world entities by removing irrelevant adjectives, stop words, and reducing excessive length (Fig. 2).

Step 1. Predicate rectification: if the action preposition belongs to the object (e.g., ”Computer networks; consist; of several assets such as Hardware”) and this preposition is directly followed by a verb, then it is added to the predicate.

Step 2. Stopword Removal and Lemmatization: Cleanse each element of the triplet by removing stopwords using a predefined list and applying lemmatization to streamline and standardize the data ensuring that no part of the triplet remains empty after this process, else the triplet is removed.

Step 3. Subject and Object Cleaning: represents the critical step, begins by checking that both the subject and object must contain a noun phrase. If this condition is met, we propose two options for cleaning, noting that we treat the subject and object independently, else the triplet is removed.

- **Option 1: Dependency Tree Analysis**

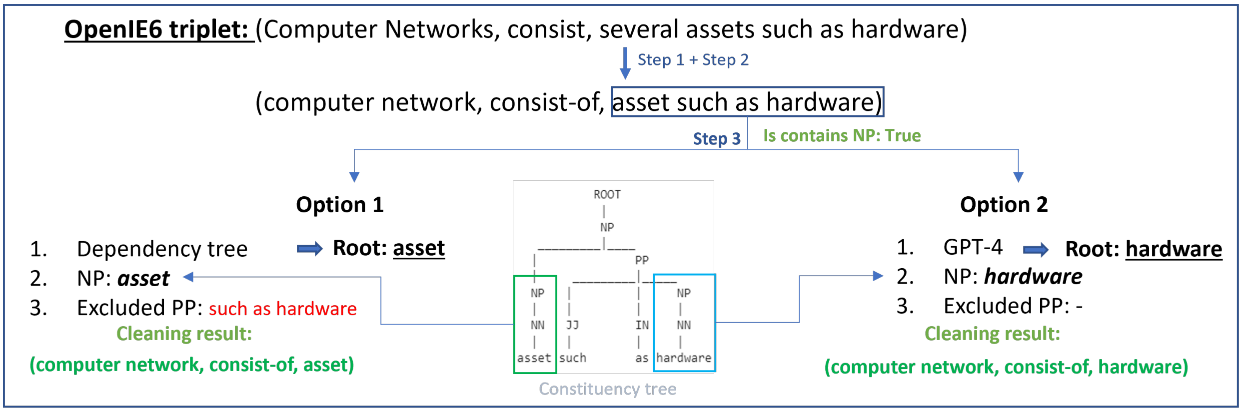

Fig. 2. Algorithm Cleaning Example

1. Determine the root (most crucial word) of the subject or object using the dependency tree from Stanford CoreNLP [3].
2. Extract the smallest noun phrase (NP) that includes this root.
3. If this NP is followed by a prepositional phrase using 'of,' a common structure in English, we append it to the extracted NP (e.g "motivation of attacker").
4. Exclude Subordinate Clauses (SBAR) and other prepositional phrases (PPs).

- **Option 2: GPT-4 Head Identification**

1. Employ GPT-4 to identify the head (most important word) of the subject or the object using a "zero-shot" approach, bypassing the need for training examples.
2. Follow steps 2, 3, and 4 of Option 1.

After cleaning the triplets from OpenIE, we merge them with the hypernym triplets.

3.2. Triplets Validator

Triplets Validator (TV) component is designed to filter out erroneous triplets resulting from the information extraction process while maintaining domain independence of the approach. To achieve this, we employ an LLM-based validation using the GPT-4 model, ensuring that the meaning of each triplet accurately reflects the source sentence. utilizing a zero-shot prompting technique, the task is simplified by converting the triplet into an affirmation, which GPT-4 then verify it's corresponding against the source sentence. Figure 3 illustrates the used prompt.

4. Evaluation

Evaluating our approach presents a challenge because of the absence of universal gold standards (GSs) for associating texts with extracted triplets without relying on a specific ontology. Since our approach is domain-independent, we employ two GSs about distinct domains (i.e., computer science and music).
Computer Science gold standard(CS-GS) enables us to compare our approach with others specifically tailored to domains like SciCero [9] and with state of the art LLM (GPT-4). Music GS allows us to evaluate the adaptability of our approach and its potential applicability across varied fields without the need for domain-specific adjustments.

[3] https://stanfordnlp.github.io/CoreNLP/depparse.html

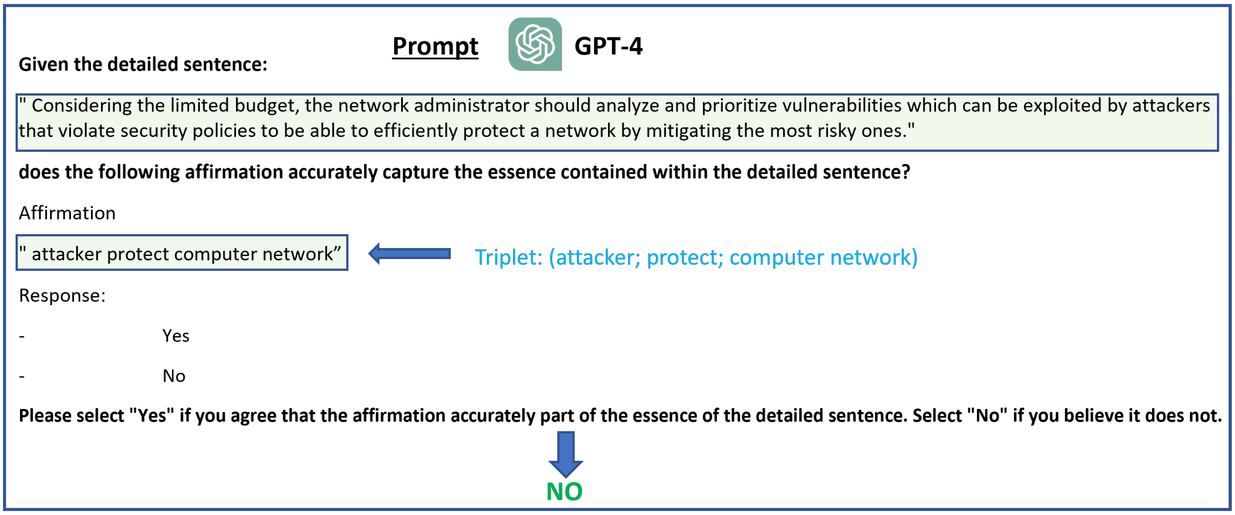

Fig. 3. Triplets Validation prompt

4.1. Gold Standard Creation

To create the CS-GS, we selected 12 abstracts of computer science articles from the Web of Science dataset provided by [15], comprising a total of 108 sentences. Abstracts were chosen based on their topics, recognizing the significant role topics play in influencing terms and writing style. Carefully selecting two abstracts for each topic, ensured content diversity. These abstracts were segmented into sentences, and corresponding triplets were extracted. Subsequently, three annotators reviewed the triplets, with inclusion determined by majority vote, resulting in a GS containing 180 triplets, including 247 unique entities and 100 unique predicates. The Music GS was derived from the corpus provided by [6]. From this corpus, we randomly selected the document titled ”20th Century Music”, comprising 100 sentences. Employing the same methodology used for creating the CS-GS, we extracted and annotated triplets. The Music GS ultimately contained 410 triplets, including 488 unique entities and 155 unique predicates.

4.2. Evaluation Method

For evaluating the performance of our approach, we conducted manual matching by three annotators, employing a majority vote to associate each extracted triplet with its equivalent in the GS. Notably, during this phase, if a triplet is deemed correct but is not present in the GS, it is added to ensure comprehensiveness. Three conventional metrics are used: Precision (P), Recall (R), and F1 scores, comparing the extracted triplets to the GS triplets.

To examine the effects of the component TV, we reconducted the evaluation after its removal from the G-T2KG pipeline. We followed the same evaluation procedure for SciCero, additionally, we conducted two experiments with ChatGPT-4:

- *”GPT-4-Exp 1”* involved using a few-shot prompting approach (four examples) while providing the paragraph as context, and then asking the model to extract triplets from a sentence.
- *”GPT-4-Exp 2”* incorporated more detailed instructions based on the errors identified in the first experiment, such as addressing coreferences and excluding pronouns from triplets, handling multiple triples in cases of conjunctions, and utilizing a set of mentions for extraction.

The results are illustrated in Table 1.

In addition, we extended our evaluation methodology to the music corpus to assess the adaptability and performance of our approach across different domains. However, it’s important to note that we focused solely on evaluating our approach rather than comparing it to other methods in the music domain. Table 2 summarizes these results.

Table 1. Evaluation results on computer science corpus

Corpus	Approaches	Precision	Recall	F1-score
Computer Science	G_T2KG_Opt1 without TV	58.50%	**47.77%**	52.59%
	G_T2KG_Opt1	72.07%	44.44%	**54.98%**
	G_T2KG_Opt2 without TV	50.00%	44.44%	47.05%
	G_T2KG_Opt2	**75.00%**	41.66%	53.56%
	SciCero	72.09%	17.22%	27.50%
	GPT-4-Exp 1	25.65%	32.77%	28.77%
	GPT-4-Exp 2	27.81%	43.88%	34.04%

Table 2. Evaluation results on music corpus

Corpus	Approaches	Precision	Recall	F1-score
Music	G_T2KG_Opt1 without TV	47.75%	36.34%	41.27%
	G_T2KG_Opt1	**64.88%**	35.12%	45.57%
	G_T2KG_Opt2 without TV	46.97%	**37.56%**	41.74%
	G_T2KG_Opt2	63.97%	36.60%	**46.56%**

4.3. Results Analysis

Experimental results shown at table 1 demonstrate in comparison to domain-specific methods, G-T2KG outperforms in recall, particularly against SciCero in the CS corpus. SciCero's lower recall, resulting from its ontology-based approach, limits extractions to domain-specific triplets, thereby enhancing its precision. Upon examining the overlap between the two approaches, it is observed that our method identifies 23% of the triplets extracted by SciCero. This is noteworthy considering that 97% of SciCero's triplets are also captured by DyGIEPP [30], which extracts predefined relations that may not be explicitly mentioned in the text. Furthermore, no SciCero triplet comes from their OpenIE [3] which underscores the importance of syntactic cleaning in our method to prevent information loss. For instance, G-T2KG detects some domain triplets missed by SciCero, such as *(network administrator; prioritize; vulnerability)* and *(domain ontology; define; concept)*.

On the other hand, compared to the triplets extracted by ChatGPT in both Experiment 1 and Experiment 2, G-T2KG also excels in both recall and precision. This is explained by the fact that the ChatGPT triplets contain common errors, such as poor handling of conjunctions leading to missed triplets, and errors arising from including "and" or "or" within a triplet, which are considered incorrect even though Experiment 2's prompt explicitly requested managing conjunctions with an example, yet the errors persisted. Additionally, the triplets extracted by GTP-4 at times result in nonsensical segmentations of sentences, such as using verbs as objects or having "with" or "to" as predicates (i.e.,*(tutorials for computer programming; can be; tedious to create)* and *(Severe consequences; such as; large blackout)*), which is consistent with findings from a [32] study indicating that GPT-4's performance is significantly weaker when dealing with a domain specific corpus.

Looking at the results shown in Table 2, the generalizability of G-T2KG is evident through consistent metrics across various domains , achieving satisfactory results in the Music corpus as well. This demonstrates the method's wide applicability in different fields without needing significant adaptation, and the possibility of extending it by using domain-specific models such as NER to target more specific entities in order to achieve results that meet specific needs.

Lastly the integration of GPT-4 as a validation tool significantly improves precision by 14% to 25%, with a minimal impact on recall (reduction of up to 3.3%), highlighting the effectiveness of LLMs in maintaining quality and relevance in KG construction while ensuring domain independence. The use of GPT-4 to verify the correspondence of triples with their source sentences is relevant, especially since there are cases where tools like OpenIE might ignore negation, for example, resulting in a completely incorrect triplet. Table 3 illustrates examples of using GPT-4 as a validation tool.

Table 3. Examples of valid and discarded triples according to the validation performed by GPT-4

Sentences	Triplets	Validator
Considering the limited budget, the network administrator should analyze and prioritize vulnerabilities which can be exploited by attackers that violate security policies to be able to efficiently protect a network by mitigating the most risky ones.	(attacker; protect; computer network)	False
Unless the developer is creating its own programming language, data structure, or describing the formal background of an existing one, he will not need to dive deep into formal languages, grammar or automaton theory.	(developer; to dive; formal language theory) and (developer; to dive; grammar)	False
In this paper, the attacker's motivation is considered in the process of security risk analysis, so network administrators are able to analyze security risks more accurately.	(attacker's motivation network administrator; be-in; process of security risk analysis)	False
Computer networks consist of several assets such as hardware, software, and data sources.	(computer network; consist-of; asset)	True

Despite the advantages and the problems our approach resolves, like others, it has certain limitations, notably its pipeline structure. Errors in one component can affect subsequent components, leading to cumulative inaccuracies. Additionally, the use of multiple separate models based on deep learning architectures can also represent a limitation in terms of execution time and resources for large corpora. Finally, the use of GPT-4 for validation makes this approach unsuitable for contexts involving confidential data, such as in medical or economic domains.

5. Conclusion and future work

In this paper, we introduced the G-T2KG pipeline for constructing knowledge graphs independently of the domain from text. This pipeline extracts triplets using two different methods: the first utilizes an OpenIE system, and the second is specifically designed for is-a relationships to enrich semantic relations. We also introduced a new syntactic cleaning component to remove noise from OpenIE triplets, based on the fact that entities are noun phrases. Additionally, we proposed a new way to validate triplets without the need for domain-specific classifiers by leveraging the capabilities of the LLM GPT-4. To evaluate our approach, we constructed gold standards in two different domains (computer science and music), which contain sentences as well as the extracted triplets. The evaluation of our approach demonstrated the crucial role of syntactic cleaning and LLM validation in improving the precision of our method, surpassing other approaches. This highlights the potential of using LLMs in the graph construction process.

In future works, we aim to utilize open-source LLMs (e.g., Llama[29]) to allow the application of our approach on confidential data and apply them at other stages of the process. We also plan to further enrich our graphs by incorporating entity typing.

Lastly, we have made the source code for G-T2KG publicly available to ensure our results are reproducible and to support the scientific community.

References

[1] Abu-Salih, B., 2021. Domain-specific knowledge graphs: A survey. Journal of Network and Computer Applications 185, 103076.

[2] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al., 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774 .

[3] Angeli, G., Johnson Premkumar, M.J., Manning, C.D., 2015. Leveraging linguistic structure for open domain information extraction, in: Zong, C., Strube, M. (Eds.), 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), Association for Computational Linguistics, Beijing, China. pp. 344–354. URL: https://aclanthology.org/P15-1034, doi:10.3115/v1/P15-1034.

[4] Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L., Auli, M., 2019. Cloze-driven pretraining of self-attention networks, in: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China. pp. 5360–5369. URL: https://aclanthology.org/D19-1539, doi:10.18653/v1/D19-1539.

[5] Beltagy, I., Lo, K., Cohan, A., 2019. SciBERT: A pretrained language model for scientific text, in: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China. pp. 3615–3620. URL: https://aclanthology.org/D19-1371, doi:10.18653/v1/D19-1371.

[6] Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., Saggion, H., 2018. SemEval-2018 task 9: Hypernym discovery. URL: https://aclanthology.org/S18-1115, doi:10.18653/v1/S18-1115.

[7] Carta, S., Giuliani, A., Piano, L., Podda, A.S., Pompianu, L., Tiddia, S.G., 2023. Iterative zero-shot llm prompting for knowledge graph construction. arXiv preprint arXiv:2307.01128 .

[8] Chen, S., Zhang, Y., Yang, Q., 2021. Multi-task learning in natural language processing: An overview. arXiv preprint arXiv:2109.09138 .

[9] Dessí, D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E., 2022. Scicero: A deep learning and nlp approach for generating scientific knowledge graphs in the computer science domain. Knowledge-Based Systems 258, 109945.

[10] Ehrlinger, L., Wöß, W., 2016. Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS) 48, 2.

[11] Ferragina, P., Scaiella, U., 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities), in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA. p. 1625–1628. URL: https://doi.org/10.1145/1871437.1871689, doi:10.1145/1871437.1871689.

[12] Hur, A., Janjua, N., Ahmed, M., 2021. A survey on state-of-the-art techniques for knowledge graphs construction and challenges ahead, in: 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 99–103. doi:10.1109/AIKE52691.2021.00021.

[13] Kertkeidkachorn, N., Ichise, R., 2018. An automatic knowledge graph creation framework from natural language text. IEICE TRANSACTIONS on Information and Systems 101, 90–98.

[14] Kolluru, K., Adlakha, V., Aggarwal, S., Mausam, Chakrabarti, S., 2020. OpenIE6: Iterative Grid Labeling and Coordination Analysis for Open Information Extraction, in: Webber, B., Cohn, T., He, Y., Liu, Y. (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online. pp. 3748–3761. URL: https://aclanthology.org/2020.emnlp-main.306, doi:10.18653/v1/2020.emnlp-main.306.

[15] Kowsari, K., Brown, D.E., Heidarysafa, M., Meimandi, K.J., Gerber, M.S., Barnes, L.E., 2017. Hdltex: Hierarchical deep learning for text classification. CoRR abs/1709.08267. URL: http://dblp.uni-trier.de/db/journals/corr/corr1709.html#abs-1709-08267.

[16] Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D., 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task, in: Proceedings of the fifteenth conference on computational natural language learning: Shared task, pp. 28–34.

[17] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D., 2014. The stanford corenlp natural language processing toolkit, in: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60.

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

[19] Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C., 2011. Dbpedia spotlight: shedding light on the web of documents, in: Proceedings of the 7th international conference on semantic systems, pp. 1–8.

[20] Mihindukulasooriya, N., Tiwari, S., Enguix, C.F., Lata, K., 2023. Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: International Semantic Web Conference, Springer. pp. 247–265.

[21] Moro, A., Raganato, A., Navigli, R., 2014. Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics 2, 231–244. URL: https://aclanthology.org/Q14-1019, doi:10.1162/tacl_a_00179.

[22] Oramas, S., Ostuni, V.C., Noia, T.D., Serra, X., Sciascio, E.D., 2016. Sound and music recommendation with knowledge graphs. ACM Transactions on Intelligent Systems and Technology (TIST) 8, 1–21.

[23] Papay, S., Klinger, R., Pado, S., 2022. Constraining linear-chain CRFs to regular languages, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=jbrgwbv8nD.

[24] Roller, S., Kiela, D., Nickel, M., 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora, in: Gurevych, I., Miyao, Y. (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia. pp. 358–363. URL: https://aclanthology.org/P18-2057, doi:10.18653/v1/P18-2057.

[25] Saha, A., Pahuja, V., Khapra, M.M., Sankaranarayanan, K., Chandar, S., 2018. Complex sequential question answering: towards learning to converse over linked question answer pairs with a knowledge graph, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press.

[26] Salatino, A.A., Osborne, F., Thanapalasingam, T., Motta, E., 2019. The cso classifier: Ontology-driven detection of research topics in scholarly articles, in: Digital Libraries for Open Knowledge: 23rd International Conference on Theory and Practice of Digital Libraries, TPDL 2019, Oslo, Norway, September 9-12, 2019, Proceedings 23, Springer. pp. 296–311.

[27] Schmitz, M., Soderland, S., Bart, R., Etzioni, O., et al., 2012. Open language learning for information extraction, in: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 523–534.

[28] Stylianou, N., Vlachava, D., Konstantinidis, I., Bassiliades, N., Peristeras, V., 2022. Doc2kg: Transforming document repositories to knowledge

graphs. International Journal on Semantic Web and Information Systems (IJSWIS) 18, 1–20.

[29] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Singh Koura, P., Lachaux, M.A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., Scialom, T., 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv e-prints , arXiv:2307.09288doi:10.48550/arXiv.2307.09288, arXiv:2307.09288.

[30] Wadden, D., Wennberg, U., Luan, Y., Hajishirzi, H., 2019. Entity, relation, and event extraction with contextualized span representations, in: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China. pp. 5784–5789. URL: https://aclanthology.org/D19-1585, doi:10.18653/v1/D19-1585.

[31] Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Chen, E., 2023. Large language models for generative information extraction: A survey. ArXiv abs/2312.17617. URL: https://api.semanticscholar.org/CorpusID:266690657.

[32] Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N., 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. ArXiv abs/2305.13168. URL: https://api.semanticscholar.org/CorpusID:258833039.