Critical Analysis of Knowledge Graph Completion Methods in the Medical Domain

Kexin Weng
Department of Computer Science
University of Warwick
Coventry, United Kingdom
Kexin.weng@warwick.ac.uk

***Abstract*—Knowledge Graph Completion (KGC) plays a key role in the extraction of drug-adverse reaction relationships, monitoring of drug interactions, and supporting clinical diagnosis in the medical field. While CodeKGC performs well in KGC tasks, its reliance on proprietary models such as Codex limits its applicability in the specialty medical field. This study proposes the use of open-source sequence generation models, such as CodeT5 and BioGPT, combined with Schema-aware Prompts to enhance the performance of medical KGC tasks. We address the shortcomings of existing evaluation methods, apply Schema-aware Prompts to improve the generation of complex data, and evaluate our approach on medical datasets such as ADE_Corpus_V2 and BC5CDR, demonstrating improvements in adaptability and effectiveness.**

***Index Terms*—Knowledge Graph Completion, Medical Knowledge Graph, Schema-aware Prompt, CodeT5, BioGPT**

I. Introduction

A. Research Background

Knowledge Graph Completion (KGC) has become increasingly significant in the medical field, particularly in tasks like drug-adverse effect relationship extraction, drug interaction monitoring, and clinical diagnosis support. Medical Knowledge Graphs (MKGs), as a structured knowledge representation, play a critical role in clinical information extraction, drug repurposing, and diagnosis assistance. For example, by automatically extracting drug-adverse effect relationships, KGC supports drug safety monitoring, helping to identify potential risks and improve patient safety and medical decision-making [1].

B. Research Motivation

CodeKGC has demonstrated good performance in KGC tasks [2]. However, its reliance on proprietary models like Codex [3] limits its adaptability and reproducibility in the medical domain. Although Codex performs well in general tasks, it is often less effective when dealing with specialized medical data compared to optimized open-source alternatives. This limitation highlights the need for exploring more accessible model frameworks and better schema-aware prompting techniques tailored to medical data structures. Therefore, this study adopts smaller sequence generation models such as CodeT5 and BioGPT [4] as open-source foundations, combined with schema-aware prompting, to enhance performance and adaptability in medical KGC tasks. This research focuses on: (1) To address the limitations of current evaluation methods. (2) To propose schema-aware prompting for better generation of complex data. (3) To evaluate sequence generation models on multiple medical datasets, such as ADE_Corpus_V2 [5] and BC5CDR [6], validating their effectiveness in tasks like drug-adverse effect extraction and biomedical entity recognition.

II. Related Work

Knowledge Graphs (KGs) consist of nodes (entities) and edges (relations), representing structured information about the real world. Knowledge Graph Completion (KGC) aims to populate KGs with new entities and relations extracted from unstructured text [7], [8]. Traditional KGC methods rely on pipelines involving tasks such as Named Entity Recognition (NER), Entity Linking, and Relation Extraction [8]. These tasks are performed sequentially using independent modules to construct the KG.

With advances in deep learning, end-to-end frameworks have gained attention. These frameworks redefine KGC as a natural language generation problem, leveraging pre-trained language models (LLMs) to generate structured text containing entities and relations.

For example, CasRel (Cascade Binary Tagging Framework) [7] was designed to extract relational triples (subject, relation, object) from unstructured text. Its key features include subject tagging, relation-specific tagging, and relation-object tagging. CasRel first identifies all potential subject entities in a sentence. For each subject, it sequentially predicts possible relations and extracts corresponding objects using binary tagging. This step-by-step approach simplifies complex tasks and makes them more controllable.

This approach treats KGC as a natural language generation task, converting relational structures into serialized text and leveraging LLMs to make predictions. These generative models have shown success in tasks such as entity extraction, relation extraction, and event extraction. However, generative models still face challenges when handling complex structured tasks, especially in overlapping information and topological structure dependencies [9].

LLMs are primarily pre-trained on free-form text. When dealing with serialized structured data (e.g., flattening a graph into a string), the differences from training data make it difficult for the model to capture interdependencies [10]. This often requires large amounts of task-specific training data, and the generated results may contain structural errors or semantic inconsistencies, further limiting the effectiveness of KGC.

ZeroRTE [11] makes a significant contribution by extending relation triplet extraction to the zero-shot setting. It introduces the RelationPrompt framework to address data scarcity. RelationPrompt utilizes language models to generate synthetic relation samples and designs a Triplet Search Decoding method, enabling the extraction of multiple triplets from a single sentence. On both the FewRel and Wiki ZSL datasets, this method is shown to outperform existing zero-shot relation classification methods and verifies the effectiveness of synthetic data and multi-triplet decoding for zero-shot triplet extraction.

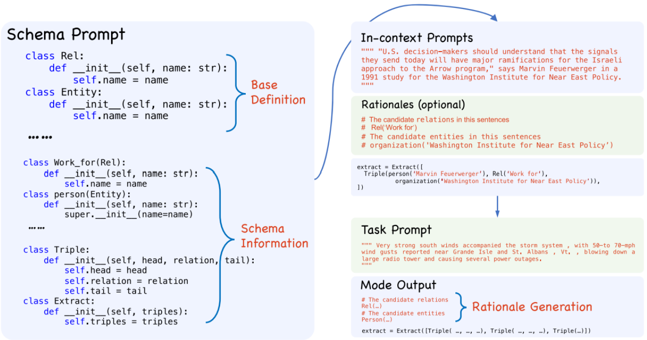
Moreover, recent research has shown that code-based language models (Code-LLMs) have exceptionally strong reasoning over structured data (e.g., program generation, code completion) [12]. These models perform well at modeling complex structural dependencies thanks to training on large amounts of code. Code-LLMs redefine KGC as a code generation task and incorporate explicit structural modeling via their structured data reasoning capabilities.

Expanding upon these methods, CodeKGC proposes a new method that parses natural language inputs into code formats as opposed to the more traditional serialized text [2] (Figure 1). Innovations include schema-aware code prompts, encoding KG patterns as structured code formats to maintain semantic formats, and enhanced generative methods, utilizing rationale-augmented generation to extract entities and relations efficiently [13].

Using ADE, CONLL04, and SciERC datasets, their experimentally show that CodeKGC outperforms baselines in zero-shot as well as few-shot settings, especially when dealing with the challenge of overlapping information [2].

There are multiple strategies to prompt engineering in LLMs, including schema-aware prompting and example-based prompting. Schema-aware prompting provides data structures that are specific to certain types of data, and task patterns which better explain how those structures map to the core task meaning. Example-based prompting gives a few shots of context to boost performance in low-resource settings. For example, a study [13] proposes a ”Reference As Prompt (RAP)” method that uses schema-aware references and similar examples as prompts and achieves significant performance improvement on KGC tasks in low-resource scenarios.

CodeKGC presents a schema-aware code prompt where entities and relations in the coded form are presented to the model to capture structural and semantic information. This prompt engineering strategy is not only compatible with the structural requirements of the task but also takes advantage of the reasoning power of code language models [2].



```
class Rel:
    def __init__(self, name: str):
        self.name = name
class Entity:
    def __init__(self, name: str):
        self.name = name

......

class Work_for(Rel):
    def __init__(self, name: str):
        self.name = name
class person(Entity):
    def __init__(self, name: str):
        super.__init__(name=name)
......

class Triple:
    def __init__(self, head, relation, tail):
        self.head = head
        self.relation = relation
        self.tail = tail
class Extract:
    def __init__(self, triples):
        self.triples = triples
```

```
""" "U.S. decision-makers should understand that the signals
they send today will have major ramifications for the Israeli
approach to the Arrow program," says Marvin Feuerwerger in a
1991 study for the Washington Institute for Near East Policy.
"""
```

```
# The candidate relations in this sentences
#  Rel('Work for')
# The candidate entities in this sentences
# organization('Washington Institute for Near East Policy')
```

```
extract = Extract([
  Triple(person('Marvin Feuerwerger'), Rel('Work for'),
           organization('Washington Institute for Near East Policy')),
])
```

```
""" Very strong south winds accompanied the storm system , with 50-to 70-mph
wind gusts reported near Grande Isle and St. Albans , Vt. , blowing down a
large radio tower and causing several power outages.
"""
```

```
# The candidate relations
Rel(…)
# The candidate entities
Person(…)
extract = Extract([Triple( …, …, …), Triple( …, …, …), Triple(…)])
```

Fig. 1. CodeKGC converts natural language inputs into code format.

III. Critical Analysis

A. *Limitations of CodeKGC's Evaluation Methods*

The methods used for evaluating CodeKGC have their limitations. First, the paper primarily uses the F1 score as the principal evaluation metric. The F1 score, which quantifies the overall accuracy of extracted triplets, however, does not take into account what entity categories contribute most to the performance of the task. The paper specifically declares that any triplet predictions are correct if and only if all head and tail entities and relations exactly match that one. However, the F1 score does not account for entity category information. It also means that even if the model predicts all proper positions and relations of entities correctly, its predictability of entity categories' performance may not be satisfactorily reflected. Furthermore, instead of performing specific error case analysis, CodeKGC does this by running multiple experiments and averaging the results, meaning insights as to where the model can get significantly better are more limited.

These limitations reflect deficiencies in evaluating methods for information extraction (IE) in a broader sense. Current evaluation frameworks overly rely on the accuracy of surface matching and neglect more in-depth usage of semantic information. For instance, while current evaluation metrics are often found to be insufficient in identifying semantically equivalent results, this leads to an underestimate of model performance in real-world applications [14].

B. *Adaptability to Complex Graph Structures*

Although CodeKGC improves the understanding of knowledge graph structures through schema-aware prompts, it still faces challenges in adapting to complex graph structures. CodeKGC is mainly optimized for regular structured data and may struggle with highly dynamic or irregular data, such as real-time data streams or cross-domain graphs. Additionally, when handling complex scenarios involving multiple entities and overlapping relationships, generative large models often require multi-step reasoning, which significantly increases computational cost and may reduce real-time applicability.

C. *Performance Limitation in Domain-Specific Scenarios*

Experimental results show that CodeKGC exhibits notable performance limitations in tasks within specific domains, such as medicine. This may be due to Codex's pretraining, which is primarily based on general-purpose code data and lacks support for domain-specific terms and knowledge. Additionally, the design of schema-aware prompts does not fully utilize the structures of domain-specific knowledge graphs, limiting the model's ability to adapt to complex domain data.

IV. Future Work

A. Improved Evaluation Methods

To address the limitations of current evaluation methods, future research could design more fine-grained and semantically aware evaluation frameworks. For example, accuracy could be measured separately for subjects (head), predicates (relation), and objects (tail) in knowledge graphs to better reveal model performance across different dimensions. Additionally, introducing semantic similarity scoring for triplets with similar meanings but different expressions could more accurately assess model generalization capabilities. Such approaches would not only enhance the fairness of evaluations but also provide clear directions for model improvements.

B. Domain-Specific Fine-Tuning on Diverse Medical Datasets

To overcome performance bottlenecks in domain-specific tasks, future research could explore fine-tuning models on diverse medical datasets [15]. By incorporating more medical-related data, such as disease-drug relationships and drug side effects, the model's understanding of medical terms and semantics could be further optimized. Collaborating with medical experts to design more precise and comprehensive schema-aware prompts could also improve the model's adaptability to medical knowledge graph construction tasks.

C. Adaptability to Complex Graph Structures

To enhance adaptability to complex graph structures, future work could introduce dynamic prompt generation mechanisms that adjust prompt content automatically based on real-time data changes. For example, for dynamic or heterogeneous graphs, tools like graph neural networks (GNNs) or other structural modeling techniques could be integrated with CodeKGC's generative capabilities to improve understanding and processing of complex scenarios. Additionally, optimizing inference algorithms to reduce the computational cost of multi-step generation would also be a critical area for improvement.

V. Conclusion

This study analyzes the contributions and limitations of the CodeKGC method in knowledge graph completion tasks. While CodeKGC introduces innovative schema-aware code prompts and generative methods that significantly enhance the understanding of knowledge graph structures, its limitations in evaluation methods, performance bottlenecks in domain-specific scenarios, and challenges in adapting to complex graph structures restrict its broader applicability.

To address these challenges, future research can focus on designing more fine-grained and semantically aware evaluation frameworks to improve performance across diverse tasks; leveraging diverse medical datasets and domain expert knowledge to enhance adaptability in medical knowledge graph construction; and introducing dynamic prompts and optimized inference mechanisms to improve modeling of complex graphs. These directions could not only enhance CodeKGC but also provide more comprehensive and efficient solutions for knowledge graph construction tasks.

References

[1] L. Murali, G. Gopakumar, D. M. Viswanathan, and P. Nedungadi, "Towards electronic health record-based medical knowledge graph construction, completion, and applications: A literature study," *Journal of biomedical informatics*, vol. 143, p. 104403, 2023.

[2] Z. Bi, J. Chen, Y. Jiang, F. Xiong, W. Guo, H. Chen, and N. Zhang, "Codekgc: Code language model for generative knowledge graph construction," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 3, pp. 1–16, 2024.

[3] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[4] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, "Biogpt: generative pre-trained transformer for biomedical text generation and mining," *Briefings in bioinformatics*, vol. 23, no. 6, p. bbac409, 2022.

[5] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of biomedical informatics*, vol. 45, no. 5, pp. 885–892, 2012.

[6] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, "Biocreative v cdr task corpus: a resource for chemical disease relation extraction," *Database*, vol. 2016, 2016.

[7] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," *arXiv preprint arXiv:1909.03227*, 2019.

[8] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," *arXiv preprint arXiv:2010.12812*, 2020.

[9] D. Rajagopal, A. Madaan, N. Tandon, Y. Yang, S. Prabhumoye, A. Ravichander, P. Clark, and E. Hovy, "Curie: An iterative querying approach for reasoning about situations," *arXiv preprint arXiv:2104.00814*, 2021.

[10] A. Madaan and Y. Yang, "Neural language modeling for contextualized temporal graph generation," *arXiv preprint arXiv:2010.10077*, 2020.

[11] Y. K. Chia, L. Bing, S. Poria, and L. Si, "Relationprompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction," *arXiv preprint arXiv:2203.09101*, 2022.

[12] A. Madaan, S. Zhou, U. Alon, Y. Yang, and G. Neubig, "Language models of code are few-shot commonsense learners," *arXiv preprint arXiv:2210.07128*, 2022.

[13] Y. Yao, S. Mao, N. Zhang, X. Chen, S. Deng, X. Chen, and H. Chen, "Schema-aware reference as prompt improves data-efficient knowledge graph construction," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 911–921.

[14] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Named entity recognition and relation extraction: State-of-the-art," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–39, 2021.

[15] Y. Liu, X. Tian, Z. Sun, and W. Hu, "Finetuning generative large language models with discrimination instructions for knowledge graph completion," *arXiv preprint arXiv:2407.16127*, 2024.