KFC-QR: A Knowledge Fusion Constraint Method Based on Query Retrieval for CPE prediction

Jinrui Zhang[1], Linyi Han[1] and Xiaowang Zhang[1,*]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China

Abstract

Textual vulnerability description (TVD) is the record of software vulnerability by software engineers. Software engineers use the common platform enumerations (CPE) in TVDs to find software related to the vulnerability record. However, CPE is always incomplete, which makes it difficult to comprehensively identify the software affected by the vulnerability. Existing work focuses on completing CPE through CPE prediction based on the Knowledge Graph Embedding (KGE) model. However, the KGE model cannot capture the potential connections between softwares in TVD. In this poster, we propose a knowledge fusion constraint method based on query retrieval. Firstly, we extract the subgraph related to TVDs from the graph, which contains known CPEs and vulnerability types. Then, we use a large language model (LLM) combined with the subgraph to rerank the candidate CPEs predicted by the KGE model. Experiments show that our framework improves the accuracy of TVD-CPE link prediction, providing valuable support for software developers in product security assessment.

Keywords

CPE Prediction, Vulnerability Knowledge Graph, Large Language Model

1. Introduction

Textual vulnerability description (TVD) is the record of software vulnerability by software engineers. Common platform enumeration[1] (CPE) is a standardized naming convention in TVD for uniquely identifying software, hardware, and applications. Software engineers can find out the affected products through CPE in TVD. However, a TVD corresponds to multiple CPEs, and it takes several weeks to find all CPEs in the TVD. Therefore, it is essential to predict undiscovered CPEs in the TVD. Shi et al.[2] construct a vulnerability knowledge graph and use the TransE[3] model to predict the link between TVD and CPE. ULTRA[4] is an approach for learning universal and transferable graph representations. It builds relational representations as a function conditioned on their interactions, which allows a pre-trained ULTRA model to inductively generalize to unseen KGs. Alfasi et al.[5] propose the VulnScopper framework, which uses a large language model(LLM) to represent TVD and then incorporate the vulnerability embeddings into the ULTRA model for training. However, they do not consider the relationship between softwares affected in TVD. We observe that CPEs in the same TVD usually have similar software architectures or vulnerability types. In this poster, we construct a **k**nowledge **f**usion **c**onstrained framework based on **q**uery **r**etrieve (KFC-QR) to predict CPEs. Given a TVD, firstly, we generate candidate CPEs through the KGE model. Secondly, we retrieve

Posters, Demos, and Industry Tracks at ISWC 2024, November 13–15, 2024, Baltimore, USA

*Corresponding author.

jinrui_zhang@tju.edu.cn (J. Zhang); hanly2@tju.edu.cn (L. Han); xiaowangzhang@tju.edu.cn (X. Zhang)

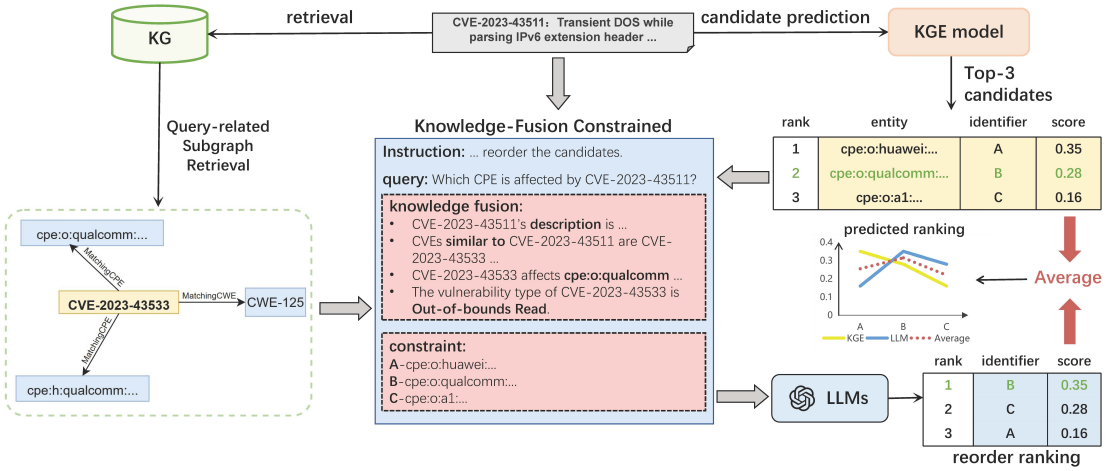

Figure 1: Overview of the KFC-QR framework.

subgraphs of similar CVEs from the vulnerability knowledge graph, which contains known CPEs and vulnerability types. Thirdly, we use LLM for re-ranking CPEs and average the scores with those output by the KGE model to obtain the final prediction results. Experiments show that our framework improves the accuracy of TVD-CPE link prediction.

2. Approach

2.1. Problem Definition

Our task is to link prediction between TVD and CPE. Formally, the vulnerability knowledge graph can be represented as G = {E, R}, where E and R represent the set of entities and relations in G, respectively. Given a TVD x_i, the link prediction task ranks all entities as $E_c = (e_1, e_2, ..., e_n)$ by calculating their scores as $S_c = (s_1, s_2, ..., s_n)$.

2.2. Approach Overview

Our approach consists of the KGE model, the query-related subgraph retrieval module and the knowledge fusion constrained inference module. We use ULTRA as the KGE model, which allows KFC-QR to link predictions for unseen entities. We use the KGE model to predict candidate CPEs for the given TVD. The Query-related Subgraph Retrieval module is used to retrieve CVE subgraphs that are similar to the given TVD, providing query-related knowledge. The Knowledge Fusion Constrained Inference module is used to fuse candidate CPEs, query related knowledge into a prompt and instruct the LLM to rerank the candidate CPEs.

2.3. Query-related Subgraph Retrieval

This module is used to retrieve the subgraph for a given TVD, providing known CPEs and vulnerability types. In the transductive setting, we directly use the TVD to query the vulnerability

Table 1
Link-prediction of TVD to CPE. ChatGPT 4 MRR is marked with "X" since it returns only the top 10 results.

Setting	Model	MRR	Hits@1	Hits@3	Hits@10
inductive	TransE	0.223	0.116	0.230	0.387
	ChatGPT 4	X	0.120	0.145	0.183
	VulnScopper(SOTA)	0.573	0.535	0.594	0.680
	Ours	**0.623**	**0.584**	**0.653**	**0.706**
	Ours KGE(ULTRA)	0.531	0.492	0.556	0.637
	Ours w/o Subgraph	0.598	0.557	0.633	0.688
transductive	TransE	0.310	0.254	0.380	0.455
	ChatGPT 4	X	0.127	0.157	0.180
	VulnScopper(SOTA)	0.651	0.533	0.706	0.768
	Ours	**0.693**	**0.580**	**0.744**	**0.780**
	Ours KGE(ULTRA)	0.553	0.483	0.577	0.683
	Ours w/o Subgraph	0.673	0.550	0.714	0.771

knowledge graph to get the subgraph. In the inductive setting, we retrieve the TVDs in the vulnerability knowledge graph that are similar to the given TVD and use the subgraphs of similar TVDs to provide knowledge for inference. We calculate the cosine similarity between TVD embeddings to obtain candidate similar TVDs. Then we extract the affected products of the candidate TVDs and filter the final similar TVDs by rules.

2.4. Knowledge Fusion Constrained Inference

Due to the natural language understanding capability of the LLM, we construct the vulnerability knowledge as prompts and instruct the LLM to rerank the candidate CPEs. We convert the triplet information of the TVD subgraph into natural language format using predefined rules, which allows LLMs to understand the knowledge more accurately. Additionally, we replace the original abstract symbols with the actual meanings of the common weakness enumeration[6]. We replace complex CPEs with simple letters to reduce the omission and fabrication problems in LLM generation.

3. Experiments

We retrieve available vulnerability knowledge from NVD[7] API and construct it into a vulnerability knowledge graph following the method of Shi et al. Our dataset has a total of 653,319 triples, of which 309,139 are TVD-CPE triples. In the transductive setup, we use 4,971 TVD-CPE triples as the test set and ensure that the entities in the test set have appeared in the training set. In the inductive setup, we divide the graph temporally, the training set contains triples up to October 1st, 2023. The triples from October 1st, 2023 to April 17th, 2024 is used as the test set. We use MRR and Hits@K as evaluation metrics. Table 1 presents the inductive and transductive link prediction results on the test set. The evaluation results indicate that *KFC-QR* performs

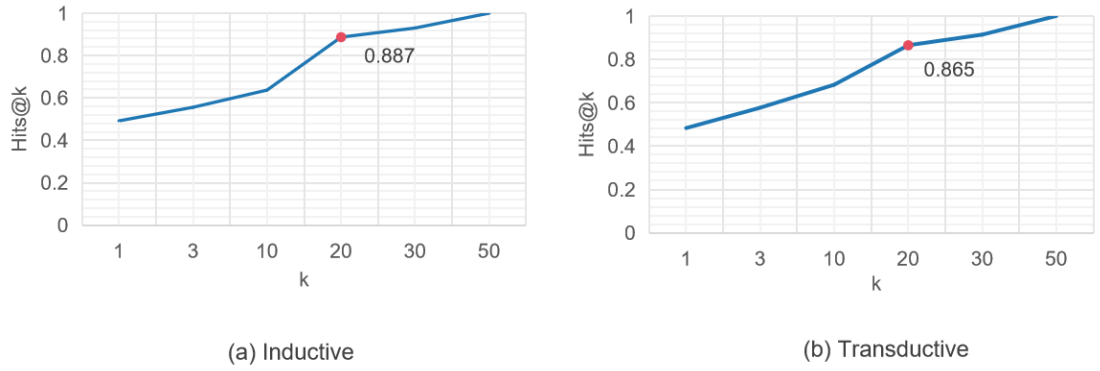

Figure 2: Hits@k results of ULTRA model.

better than all the other methods in predicting the link between CPE and TVD. We also conduct ablation experiments to validate the effectiveness of the query-related subgraph module.

4. Limitation

Our model relies on the performance of the KGE model because we rerank the top K candidate CPEs predicted by the KGE model. As shown in Figure 2, considering the accuracy of the model and the size of the candidate CPEs, we set k=20. We rerank the top 20 candidate CPEs predicted by the KGE model. In the application of other datasets, it is necessary to dynamically adjust the K value according to the performance of the model to make the candidate CPEs cover the correct CPEs as much as possible.

5. Conclusion

In this poster, we propose KFC-QR for link predictions between TVD and CPE, which considers the similarity between CPEs affected by the same TVD. Experiments demonstrate that the framework improves the link prediction accuracy between TVDs and CPEs. In the future, we plan to extend this framework to link prediction for TVD and common weakness enumeration to verify generalization.

Acknowledgement

This work was supported by the Project of Science and Technology Research and Development Plan of China Railway Corporation (N2023J044).

References

[1] MITRE, Official common platform enumeration (cpe) dictionary, 2024. URL: https://nvd.nist.gov/products/cpe.

[2] Z. Shi, N. Matyunin, K. Graffi, D. Starobinski, Uncovering cwe-cve-cpe relations with threat knowledge graphs, ACM Transactions on Privacy and Security 27 (2024) 1–26.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[4] M. Galkin, X. Yuan, H. Mostafa, J. Tang, Z. Zhu, Towards foundation models for knowledge graph reasoning, in: The Twelfth International Conference on Learning Representations, 2024.

[5] D. Alfasi, T. Shapira, A. B. Barr, Unveiling hidden links between unseen security entities, arXiv preprint arXiv:2403.02014 (2024).

[6] MITRE, Common weakness enumeration (cwe), 2024. URL: https://cwe.mitre.org.

[7] NVD, National vulnerability database (nvd), 2024. URL: https://nvd.nist.gov/general.