

LNCS 14332

Xiangyu Song · Ruyi Feng ·
Yunliang Chen · Jianxin Li ·
Geyong Min (Eds.)

Web and Big Data

7th International Joint Conference, APWeb-WAIM 2023
Wuhan, China, October 6–8, 2023
Proceedings, Part II

2
Part II



Springer

Founding Editors

Gerhard Goos

Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*

Wen Gao, *Peking University, Beijing, China*

Bernhard Steffen , *TU Dortmund University, Dortmund, Germany*

Moti Yung , *Columbia University, New York, NY, USA*

Xiangyu Song · Ruyi Feng · Yunliang Chen ·
Jianxin Li · Geyong Min
Editors

Web and Big Data

7th International Joint Conference, APWeb-WAIM 2023
Wuhan, China, October 6–8, 2023
Proceedings, Part II

Editors

Xiangyu Song  Peng Cheng Laboratory
Shenzhen, China

Yunliang Chen  China University of Geosciences
Wuhan, China

Geyong Min  University of Exeter
Exeter, UK

Ruyi Feng  China University of Geosciences
Wuhan, China

Jianxin Li  Deakin University
Burwood, VIC, Australia

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-981-97-2389-8

ISBN 978-981-97-2390-4 (eBook)

<https://doi.org/10.1007/978-981-97-2390-4>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.



Entity Alignment Based on Multi-view Interaction Model in Vulnerability Knowledge Graphs

Jin Jiang and Mohan Li^(✉)

Cyberspace Institute of Advance Technology, Guangzhou University, Guangzhou,
China

csjiangjin@e.gzhu.edu.cn, limohan@gzhu.edu.cn

Abstract. Entity alignment (EA) aims to match the same entities in different Knowledge Graphs (KGs), which is a critical task in KG fusion. EA has recently attracted the attention of many researchers, but the performance of general methods on KGs in some professional fields is not satisfactory. Vulnerability KG is a kind of KG that stores vulnerability knowledge. The text and structure information is not the same as the general KG, so the EA task faces unique challenges. First, although some vulnerabilities have a unified CVE number, in reality, the CVE number attribute value of many vulnerability entities in KG is missing. Second, vulnerability KGs often contain a large number of 1–n and n–1 relations, and general entity embedding methods may generate similar vector representations for a large number of non-identical vulnerabilities. To address the above challenges, we propose a multi-view text-graph interaction model (TG-INT) for the EA task in vulnerability KG. We use cross-lingual BERT to learn text embeddings and an optimized model called QuatAE to embed two graphs into a unified vector space. After that, we employed a multi-view interactive modeling scheme for the EA task. On the vulnerability KGs built on the vulnerability database CNNVD and CNVD, we verified the effectiveness of TG-INT. The results show that our model is not only suitable for vulnerability KGs but also achieves promising results in general KGs.

Keywords: Vulnerability knowledge graph · Entity alignment · Graph embedding · Security database

1 Introduction

The quality of a knowledge graph (KG) is essential for most downstream tasks, including automatic question-answering [1] and anomaly detection [2]. However, ensuring high-quality information in a KG is challenging and can be costly. Therefore, fusing KGs of similar fields has become attractive and valuable. Entity alignment (EA) aims to identify pairs of entities that represent the same thing across different KGs, but this can be difficult due to the heterogeneity of KGs from different sources. EA is an even more challenging task for vulnerability KGs in cyberspace security, which consist of information about vulnerabilities and

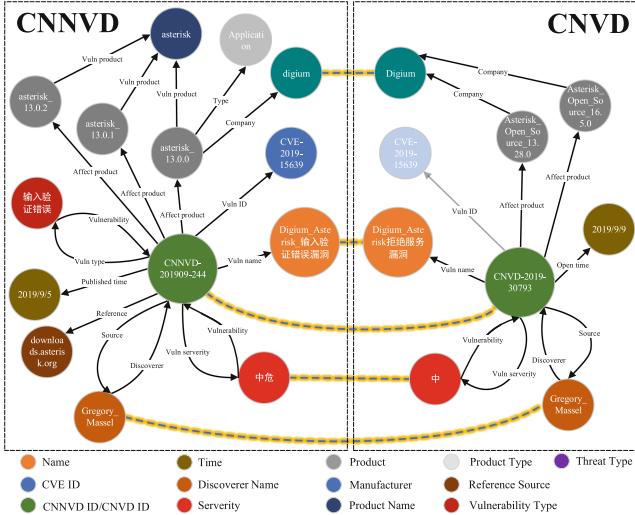


Fig. 1. An example of vulnerability KG alignment, where the alignment with missing CVE numbers is relatively difficult. (Dashed line connecting aligned entity pairs)

their associated levels of danger, products, manufacturers, etc. Unlike general KGs, vulnerability KGs contain a significant number of similar but not identical entities, such as different products from the same manufacturer and different vulnerabilities of the same type. As shown in Fig. 1, trained professionals are required to determine whether an entity pair is aligned when the CVE number is missing. Despite a large number of research on general KGs, there has been limited focus on entity alignment for vulnerability KGs.

Textual rules and logical reasoning [3,4] were the primary methods used for EA in general KGs, but these methods were not effective for cross-linguistic KGs. With the development of deep learning, geometric transformations [5,6] and graph neural networks [7–9] have been used to extract structural features for EA tasks. However, only calculating the similarity of graph structure features cannot achieve satisfactory accuracy. If text information such as entity names and descriptions are also considered, the accuracy will be greatly improved [10]. In recent studies, BERT-INT [11] constructs a BERT-based interaction model using only text information. Other models, such as UPLR [12], and SelfKG [13], combine text embedding into graph neural networks to build EA models.

In vulnerability KGs, there are many entities with similar names and some vulnerability databases mix different languages. For example, Chinese and English are both present in China National Vulnerability Database of Information Security (CNNVD) and China National Vulnerability Database (CNVD). Only relying on text similarity without considering the graph structure would not work well. However, relying only on graph embeddings is also insufficient. Due to the limited variety of relations in the vulnerability KG compared to a general KG, the semantic information contained in the graph structure is also limited. These challenges make the task of EA in vulnerability KGs difficult.

To jointly exploit text and graph structure information for EA in vulnerability KGs, we propose a multi-view **Text-Graph INTeraction** model (TG-INT). First, we design QuatAE, a QuatE-based model that models symmetric and inverse relations and learns the semantic information of entities in the graph structure. Second, we employ a cross-language BERT to acquire text information about entities. Based on the acquired textual and structural features, we build a multi-view interaction model for EA in vulnerability KGs. Our model combines similarities in three types of views, i.e., text-graph view, neighbor view and attribute view. In the text-graph and neighbor views, we consider both the graph embedding and text embedding of entities, and only extracting text embeddings in the attribute view. Experimental results demonstrate that our approach is effective not only for vulnerability KGs but also improves general cross-linguistic KG alignment. To the best of our knowledge, this is the first study to address the vulnerability KG EA task. The contributions of this paper are as follows:

- We propose **QuatAE**, a graph embedding method for EA task of vulnerability KGs. QuatAE is an efficient approach for capturing the graph structure and effectively assists with EA task.
- We design **TG-INT**, a multi-view text-graph interaction model to improve the precision of EA. Our approach integrates information from the text-graph view, the neighbor view, and the attribute view.
- We experimentally verify that TG-INT not only achieves the best results on the EA task on vulnerability KGs, but also improves the EA accuracy on general KGs.¹

2 Related Work

Knowledge Graph Embeddings. The current EA models mainly use GNN-based and geometric modeling-based methods for graph embedding. GNN [7–9] learns entity embeddings by aggregating information from their neighbors, while geometric modeling-based methods represent relations as geometric transformations between entities. TransE [5] models the relations as translational vectors in Euclidean space. QuatE [6] models the relation as a rotation in Quaternion space. These graph embedding methods capture important information about the graph structure that can be used for effective entity alignment.

EA Based on Graph Embeddings and Auxiliary Information. The initial EA models relied on graph embeddings, while MTransE [14] employs entity alignment seeds to train a transformation matrix. Later models such as IPTransE [15] and BootEA [16] have introduced iterative approaches to incorporate additional label data, while some models like NAEA [17] and AliNet [18] have integrated information about the entity’s neighbors.

¹ The code is available at <https://github.com/krypros/TG-INT>.

Recent research has been focused on entity alignment using both graph structure information and side information. JAPE [19], GCN-Align [21] and MultiKE [20] assign structural and attribute features to entities. RDGCN [22] constructs a primal graph and dual relation graph to learn complex relational information for alignment. BERT-INT [11] learns text information using only BERT to model entity, neighbor, and attribute interactions. RPR-RHGT [23] aggregates relation embeddings and path embeddings using relation-aware Transform. UPLR [12] generates pseudo-labels using semantic information from the text and learns entity embedding using the GAT network. These methods incorporate side information from different perspectives to support EA. Although most of these methods incorporate text information into the graph neural network, the process of graph aggregation may weaken the textual features of the entities themselves. Therefore, we make all relevant text, graph, relation, and attribute information available for the EA task of vulnerability KGs.

3 Problem Formulation

Knowledge Graph. A knowledge graph can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{V}, \mathcal{T})$, which consists of relation triples and attribute triples. \mathcal{E} , \mathcal{R} , \mathcal{A} and \mathcal{V} denote the set of entities, relations, attributes and attribute values, respectively. $\mathcal{T} = \mathcal{T}_r \cup \mathcal{T}_a$ denotes the set of triples, where \mathcal{T}_r is the set of relation triples and \mathcal{T}_a is the set of attribute triples. A relation triple (h, r, t) consists of entities and relation. A attribute triple (h, a, v) consists of entity, attribute and attribute values. Entities tend to have multiple meanings, and attribute values are relatively more certain.

Entity Alignment. Given two graphs $\mathcal{G}_1 = (\mathcal{E}_1, \mathcal{R}_1, \mathcal{A}_1, \mathcal{V}_1, \mathcal{T}_1)$, $\mathcal{G}_2 = (\mathcal{E}_2, \mathcal{R}_2, \mathcal{A}_2, \mathcal{V}_2, \mathcal{T}_2)$ and few entity alignment seeds $\mathcal{S}_d = \{e, e' | e \equiv e', e \in \mathcal{E}_1, e' \in \mathcal{E}_2\}$. \equiv indicates that the entity pairs are identical. Entity alignment requires that pairs of entities with the same actual meaning be found in \mathcal{G}_1 and \mathcal{G}_2 .

The objective of vulnerability KG entity alignment is similar to that of general knowledge graphs, but it specifically addresses the challenge of aligning vulnerability entities that lack CVE numbers.

4 Methodology

In this chapter, we propose QuatAE, a KG embedding method for EA. And then combine these graph embeddings with a cross-linguistic alignment model based on BERT, utilizing a three-view interaction approach.

4.1 QuatAE

Given the large number of symmetric relations in the vulnerability KG, QuatE [6] can effectively model symmetric and inverse relations. Consequently, we have improved the QuatE model to better learn graph embeddings and enhance its performance in the EA task. First, we use the known label data $\mathcal{S}_d = \{(e, e') | e \in$

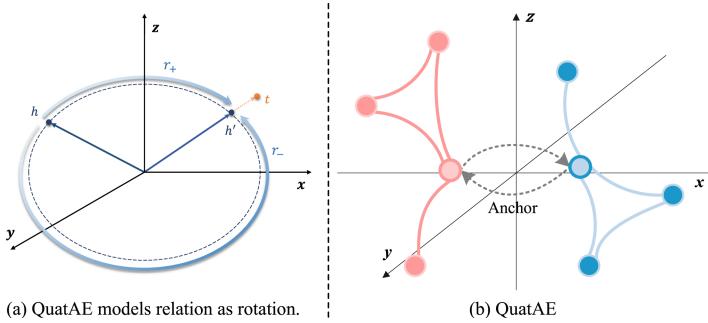


Fig. 2. QuatE and QuatAE.

$\mathcal{E}_1, e' \in \mathcal{E}_2, e \equiv e'\}$ to embed the two candidate knowledge graphs into a uniform vector space. For this purpose, we define a new relation ‘anchor’ r_a . By using \mathcal{S}_d and r_a , we can generate a set of triples $\mathcal{T}_s = \{(e, r_a, e') \cup (e', r_a, e) | (e, e') \in \mathcal{S}_d\}$ to support alignment training. The training is supervised by these alignment triples so that the same entities in both graphs are pulled in by the ‘anchor’ during the training process. As depicted in Fig. 2 (b), the ‘anchor’ acts as a bridge between the two graphs. In contrast to the direct approach of sharing the same vector between known alignment pairs, the introduction of the ‘anchor’ [24] allows identical entities to be closer together in a unified vector space, while maintaining the original structural properties of the two graphs as much as possible.

QuatAE, just like QuatE, models relation as rotational transformation in quaternion space. Thus inverse and symmetric relations can be modeled, satisfying the properties of the ‘anchor’. For a given triple (h, r, t) , QuatAE embeds both entities and relations in the quaternion space. Quaternion embeddings \mathbf{v}_h , \mathbf{v}_r and \mathbf{v}_t are represented as:

$$\begin{aligned}\mathbf{v}_h &= (v_{h,w}, v_{h,i}, v_{h,j}, v_{h,k}), \\ \mathbf{v}_r &= (v_{r,w}, v_{r,i}, v_{r,j}, v_{r,k}), \\ \mathbf{v}_t &= (v_{t,w}, v_{t,i}, v_{t,j}, v_{t,k}),\end{aligned}\tag{1}$$

where each quaternion embedding consists of a real part and three imaginary parts. Each element in the embedding is a vector. To achieve a rotation in quaternion space, we need to normalize the relation \mathbf{v}_r to the unit quaternion. This unitization likewise eliminates the effect of the embedding scale:

$$\mathbf{v}_r^{\triangleleft} = \frac{\mathbf{v}_r}{\|\mathbf{v}_r\|} = (\bar{v}_{r,w}, \bar{v}_{r,i}, \bar{v}_{r,j}, \bar{v}_{r,k}).\tag{2}$$

After obtaining the unit embedding. As shown in Fig. 2(a), QuatAE uses hamilton multiplication [6] to rotate the entity h to h' by isoclinic rotation r_+ or r_- , which makes the angle between h' and t to be zero. After the rotation, \mathbf{v}_h transforms into \mathbf{v}'_h . That the inner product of the head and tail entity embedding is zero, from which the scoring function can be obtained:

$$f(h, r, t) = \mathbf{v}_h \otimes \mathbf{v}_r^\triangleleft \bullet \mathbf{v}_t = \mathbf{v}'_h \bullet \mathbf{v}_t, \quad (3)$$

where \otimes denotes the hamilton product and \bullet denotes the element-wise multiplication. Consider that we will measure the semantic similarity of two entity embeddings by comparing their cosine distances in a subsequent alignment task. So we constrain the cosine distances of the entity embeddings in the alignment seeds to be close during the training process, and learned by minimizing the following regularized logistic loss:

$$\begin{aligned} \mathcal{L}_1 = & \sum_{(h, r, t) \in \mathcal{T} \cup \mathcal{T}_-} \log(1 + \exp(-l \cdot f(h, r, t))) \\ & + \lambda_1 \left(\|\mathbf{v}_h\|_2^2 + \|\mathbf{v}_r\|_2^2 + \|\mathbf{v}_t\|_2^2 \right) + \lambda_2 \sum_{(h, t) \in \mathcal{T}_s} \cos(\mathbf{v}_h, \mathbf{v}_t), \end{aligned} \quad (4)$$

where $l \in \{-1, 1\}$ denotes whether the triple (h, r, t) is a negative sample. \mathcal{T}_- is the set of negative sample triples generated by sampling from \mathcal{T} . We utilized the same sampling strategies as QuatE [6]. To avoid overfitting, we adopt L2-regularization and set the parameter to λ_1 . λ_2 is set to avoid the model being overly concerned with the aligned pairs. $\cos(\cdot)$ denotes the cosine distance between two entities.

QuatAE is capable of learning structure information on vulnerability KGs. By modeling entities in quaternion space, QuatAE can effectively address the widespread issue of 1–n and n–1 relations in the vulnerability KG. While alignment using QuatAE alone is superior to many graph embedding-based alignment models, it does not quite meet practical standards. To achieve optimal alignment, we combine the graph structure information learned by QuatAE with text information in the next section.

4.2 Text and Graph Interaction Model

Since the graph structure in the vulnerability KG provides limited semantic information, EA also requires considering the text-based semantic information such as entity names and descriptions. Fortunately, cross-language pre-training models like BERT can be used to learn text-based features, though the mixing of English and Chinese in CNNVD and CNVD can pose some challenges. For each entity, we use a BERT model to embed its name, description, and attribute values. We then concatenate the resulting text embeddings with the graph embeddings obtained from QuatAE and pass them through a MLP layer. Subsequently, we consider a combination of text-graph, neighbor, and attribute views, and use a unified dual aggregation function to compute matching scores between entity pairs based on information from the neighbor and attribute views. The overall framework of the model is shown in the Fig. 3.

Basic Model. Given that vulnerability data consists of both Chinese and English, pre-trained models in either language alone were weaker than cross-language

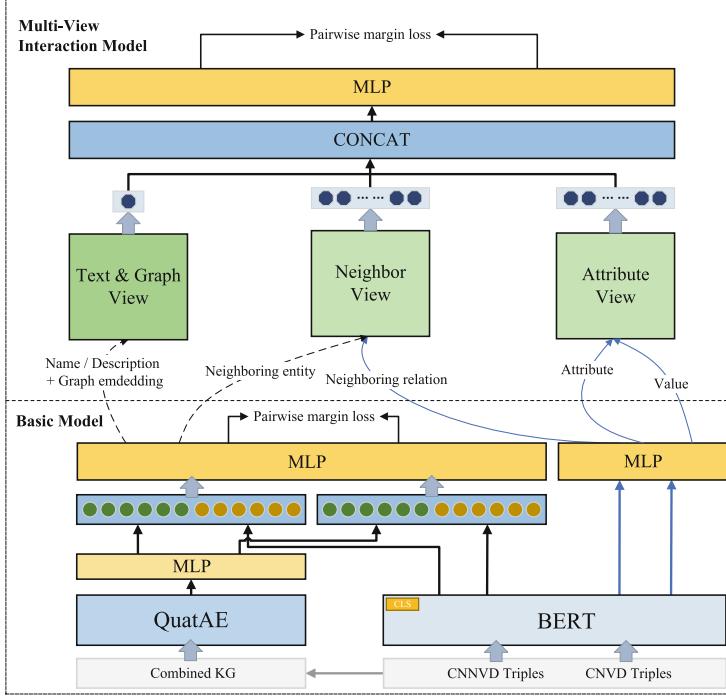


Fig. 3. The framework of TG-INT.

models. This characteristic makes our proposed solution suitable for cross linguistic KG alignment. In our scenario, we believe that relying solely on text information without taking into account the structure information would result in a loss of uniqueness of the dataset. Therefore, we designed a model that combines both text and graph information.

We first use the alignment seeds and randomly sample some negative samples $\mathcal{D} = \{(e, e_+, e_-)\}$ as training data for fine-tuning. $e_+ \in \mathcal{S}_d$ denotes the same entity as e . e_- denotes a negative sample that is not identical to e . The sampling of the negative sample is selected by the cosine similarity of the entity pairs. We take the name/description of each entity e in the training set as input to the BERT, concatenate the CLS embedding of the BERT with the QuatAE embedding, and then filter it by a MLP layer. Considering the discrepancy between the QuatAE embedding and the BERT embedding, we also pass the QuatAE embedding through a MLP layer for feature selection before concatenating:

$$G(e) = \text{MLP}(\text{QuatAE}(e)), \quad TG(e) = \text{MLP}(\text{CLS}(e) \oplus G(e)), \quad (5)$$

where \oplus denotes a concatenation between vectors. We use pairwise margin loss to fine tune the BERT:

$$\mathcal{L}_2 = \sum_{(e, e_+, e_-) \in \mathcal{D}} \max\{0, d(e, e_+) - d(e, e_-) + m\}, \quad (6)$$

where $d(\cdot)$ use l_1 distance to calculate similarity between $TG(e)$ and $TG(e_+)$ (or $TG(e_-)$). m is the margin parameter.

The approach we take in our model is similar to BERT-INT [11] in that we prioritize the description of an entity over its name since the former provides more valuable information. However, our model differs in that it considers not only the text embedding of entities but also their graph embedding. In the basic model, embeddings are provided from the text-graph view, and aligned entities can be matched by calculating the embedding similarity between entities. For a given entity pair (e_i, e_j) , its similarity score can be calculated as:

$$s_{e_i, e_j} = \frac{TG(e_i) \cdot TG(e_j)}{\|TG(e_i)\| \cdot \|TG(e_j)\|}. \quad (7)$$

To improve the alignment results, we have incorporated information from the neighbor view and the attribute view into the basic model. During the interaction model, we keep the parameters of the basic model fixed and use it as input.

Multi-view Interaction Model. We have already learned the structure and text information of the entities in the basic model. However, the neighbors and attributes of entities can still provide valuable information for EA. Entities with similar meanings usually have similar neighbors and attribute values. For this reason, we propose adding the neighbor and attribute views to interact with the text-graph view in the basic model, combining information from all views of an entity to complete the EA task.

We use a fixed number of candidate matching entity pairs from the basic model as input to the multi-view interaction model. The basic model filters out a significant number of mismatched entities and takes only a few pairs of entities for the interaction model to select. This not only saves a lot of computing time, but also makes better use of the basic model based on the text-graph view.

Our processing flow remains consistent for both the neighbor view and the attribute view. The only variation lies in the input data for each. The neighbor view requires a set of entities connected by the relation r to the target entity, while the attribute view needs a set of attribute values.

Take the neighbor view as an illustration, as shown in Fig. 4, we assist in matching entity pairs by comparing the similarity of neighboring entities. Given a pair of candidate (e_i, e_j) , their respective embedding sets of neighboring entities $\{TG(e_i)\}_{i=1}^{|\mathcal{N}(e_i)|}$ and $\{TG(e_j)\}_{j=1}^{|\mathcal{N}(e_j)|}$ are obtained from the basic model. $|\mathcal{N}(e_i)|$ and $|\mathcal{N}(e_j)|$ denote the sets of neighboring entities for entity e_i and entity e_j , respectively. The embeddings of these entities are utilized to compute a similarity matrix \mathbf{S}_r among the neighboring entities. Each element of the matrix is the similarity $s_{x,y}$ computed from Eq. (7) between the x -th neighbor of e_i and the y -th neighbor of e_j .

Meanwhile, we also consider the similarity of relations between entities and their neighbors as an influential factor in entity alignment. We use a neighboring relation similarity matrix as Mask to improve the reliability of the neighbor view. The similarity between relations is calculated similarly to the Eq. (7). It is

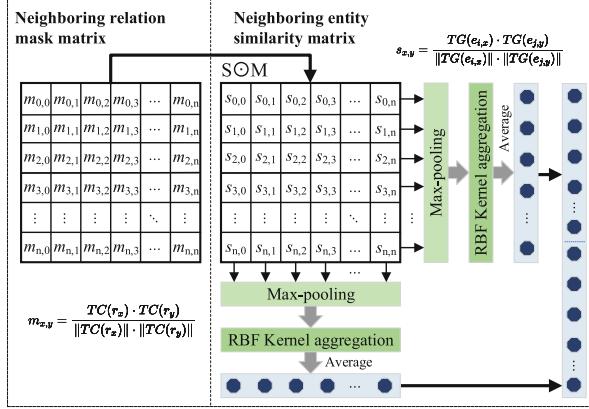


Fig. 4. Entity pair similarity matrix calculation from the neighbor view.

important to note that since the calculation of similarity between relations and attributes relies only on text similarity, as shown in Fig. 3, we only use the CLS embedding to calculate the similarity between relations/attributes:

$$TC(r) = \text{MLP}(\text{CLS}(r)), m_{x,y} = \frac{TC(r_x) \cdot TC(r_y)}{\|TC(r_x)\| \cdot \|TC(r_y)\|}, \quad (8)$$

where $m_{x,y}$ denotes the similarity between the x -th relation of e_i and the y -th relation of e_j . These relation similarities can form a matrix of relation masks \mathbf{M} . The neighboring entity similarity matrix \mathbf{S}_r and the corresponding relation mask matrix \mathbf{M} are element-wise product to obtain the final similarity matrix $\mathbf{S}'_r = \mathbf{S}_r \odot \mathbf{M}$. We used two RBF kernel aggregation functions [25] to simultaneously extract similar features from the rows and columns of \mathbf{S}'_r . To make more accurate use of similar neighbor entity pairs, we take the maximum value of each row/column of \mathbf{S}'_r as input to the RBF kernel function. That is, it only cares about how similar the most similar pairs of neighbor entities are. Thus, we can obtain the set of similarity scores for each row and column of entity e_i and entity e_j in the neighbor view \mathbf{S}'_r :

$$\varphi^N(e_i, e_j) = \left[\frac{1}{|\mathcal{N}(e_i)|} \sum_{x=1}^{|\mathcal{N}(e_i)|} \log (\text{RBF}_r(\mathbf{S}'_{r,x})) \right] \oplus \left[\frac{1}{|\mathcal{N}(e_j)|} \sum_{y=1}^{|\mathcal{N}(e_j)|} \log (\text{RBF}_c(\mathbf{S}'_{r,y})) \right], \quad (9)$$

where $\text{RBF}_r(\cdot)$ aggregates the row elements of $\mathbf{S}'_{r,x}$ and $\text{RBF}_c(\cdot)$ aggregates the column elements of $\mathbf{S}'_{r,y}$. $\mathbf{S}'_{r,x}$ is the similarity matrix after row max-pooling. $\mathbf{S}'_{r,y}$ is the similarity matrix after column max-pooling.

Similarly, in the attribute view, the input triples are replaced with the attribute triples. We can obtain the set of similarity $\varphi^A(e_i, e_j)$ for entity e_i

and entity e_j using a similar approach. Note that since the attribute values are not trained in the basic model, we only use Eq. (8) to get embeddings.

Up to now, for a given entity pair (e_i, e_j) , we combine the set of similarity scores from the text-graph, neighbor, and attribute views, concatenate them together and input them into a MLP layer to obtain the final similarity scores for the three views interactions:

$$\varphi(e_i, e_j) = \text{MLP} \left([s_{e_i, e_j} \oplus \varphi^N(e_i, e_j) \oplus \varphi^A(e_i, e_j)] \right), \quad (10)$$

Note that the MLP layer here is still optimized by the same pairwise margin loss as in Eq. (6), while the parameters of the BERT are frozen.

4.3 Entity Alignment

For an entity e in the graph \mathcal{G}_1 , We provide two options to find aligned entity in \mathcal{G}_2 . The first is EA using only the text-graph view, i.e. using only the basic model. The similarity between e and \mathcal{G}_2 is determined using Eq. (7), and the entity with the highest similarity score is chosen as the matching entity. This approach has been able to achieve superior results compared to both the text-only and the graph-only approaches, as it leverages both the graph structure information and the text information.

If a more suitable match is required, we can select the second option. The top-K candidate matching entities are quickly found by the first scheme, and then the similarity of the three views is combined using the Eq. (10). This solution utilizes a combination of graph information, text information, neighbors, and attributes to substantially enhance EA.

5 Experiments

In this chapter, we experiment with a CNNVD-CNVD vulnerability KG and three widely used cross-linguistic knowledge graphs. Details of these datasets are shown in Table 1. The experimental results demonstrate that TG-INT achieves optimal results on all datasets. Our experiments were run on a GeForce GTX 2080Ti device with 12GB of RAM.

Table 1. Datasets.

Dataset	CVD19-6K		DBP15K _{ZH_EN}		DBP15K _{JA_EN}		DBP15K _{FR_EN}	
	CNNVD	CNVD	Chinese	English	Japanese	English	French	English
#Ent	28549	12310	19388	19572	19814	19780	19661	19993
#Rel	16	10	1701	1323	1299	1153	903	1208
#Attr	4	6	8113	7173	5882	6066	4547	6422
#Rel.Tri	172047	26584	70414	95142	77214	93484	105998	105722
#Attr.Tri	9941	16986	379684	567755	354619	497230	528665	576543
#Link	2286	2286	4500	4500	4500	4500	4500	4500
#Test Link	3895	3895	10500	10500	10500	10500	10500	10500

5.1 Datasets and Implementation

Vulnerability Database and DBP15K. We obtained public vulnerability data from the China National Vulnerability Database of Information Security (CNNVD)² and the China National Vulnerability Database (CNVD)³, and defined their ontologies. For CNNVD, we defined 13 types of entities, 16 types of relations and 4 types of attributes. For CNVD, we defined 8 types of entities, 10 types of relations and 6 types of attributes. Notably, CNNVD’s ontologies mostly include all of CNVD’s ontologies, which allowed us to complement each other and obtain more comprehensive information.

Considering the limitation of GPU memory, we selected vulnerability data with shared CVE numbers between CNNVD and CNVD from September–December 2019 to create the dataset for the EA experiments. To simulate real-life situations where CVE numbers are missing, we randomly selected one-third of the data as the test set and removed the CVE number of these vulnerabilities in datasets. For vulnerabilities that have already been assigned a CVE number, we selected only the unambiguous entity pairs as seed for our EA experiments. We excluded vulnerabilities with retained CVE numbers from our test dataset since they are easier to align. Moreover, we did not include entities that can be aligned easily by text, such as time and threat types.

Despite having fewer relations and attributes than general cross-linguistic KGs, the vulnerability KG still has a sufficient number of relation triples because of the presence of similar entities. For instance, a vulnerability of a manufacturer may impact all of its products, which may share similar characteristics. Apart from the vulnerability dataset, we have conducted experiments on DBP15K datasets to test the versatility of our solution.

Parameter Settings. In our experiments, the embedding dimension of QuatAE is 800. Since QuatAE is a joint representation of an embedding by four vectors, we concatenate four 200-dimension vectors together. The hyperparameters λ_1 and λ_2 take values in [0.1, 0.2, 0.5]. The dimension of the BERT CLS embedding is 768. We use MLPs of 800 and 300 dimensions for $G(e)$ and $TG(e)$ respectively in Eq. (5). The 300-dimension MLP and 11 plus 1-dimension MLP are used in Eq. (8) and Eq. (10), respectively. In the Eq. (9), we use 20 semantic matching kernels [11]. We set both the maximum number of neighbors and the number of attributes to 50, and the number of candidates selected from the basic model of the text-graph view to 50. We found that the top-50 candidates contained 99% of the true alignment pairs in all datasets.

5.2 Experiments

Overall Performance. To validate the performance of the TG-INT model, 12 current models with high influence or good performance were selected as the

² <https://www.cnvd.org.cn>.

³ <https://www.cnnvd.org.cn>.

Table 2. Overall entity alignment experimental results. (* indicates result from our experiments)

Model	CVD19-6K*			DBP15K _{ZH_EN}			DBP15K _{JA_EN}			DBP15K _{FR_EN}		
	Hit@1	H@10	MRR	Hit@1	H@10	MRR	Hit@1	H@10	MRR	Hit@1	H@10	MRR
Only use graph structures												
MTransE (2017) [14]	0.068	0.167	0.101	0.308	0.614	0.364	0.279	0.575	0.349	0.244	0.556	0.335
IPTransE (2017) [15]	0.053	0.138	0.082	0.406	0.735	0.516	0.367	0.693	0.474	0.333	0.685	0.451
BootEA (2018) [16]	0.112	0.328	0.182	0.629	0.848	0.703	0.622	0.854	0.701	0.653	0.874	0.731
NAEA (2019) [17]	–	–	–	0.650	0.867	0.720	0.641	0.873	0.718	0.673	0.894	0.752
AliNet (2020) [18]	0.085	0.181	0.120	0.539	0.826	0.628	0.549	0.831	0.645	0.552	0.852	0.657
Combine graph structures and side information.												
JAPE (2017) [19]	0.051	0.123	0.075	0.412	0.745	0.490	0.363	0.685	0.476	0.324	0.667	0.430
GCN-Align (2018) [21]	0.077	0.246	0.132	0.413	0.744	0.549	0.399	0.745	0.546	0.373	0.745	0.532
MultiKE (2019) [20]	0.144	0.303	0.199	0.509	0.576	0.532	0.393	0.489	0.426	0.639	0.712	0.665
RDGCN (2019) [22]	0.198	0.382	0.263	0.708	0.846	0.746	0.767	0.895	0.812	0.886	0.957	0.911
BERT-INT (2020) [11]*	<u>0.743</u>	<u>0.943</u>	<u>0.814</u>	<u>0.960</u>	<u>0.982</u>	<u>0.969</u>	<u>0.959</u>	<u>0.984</u>	<u>0.970</u>	<u>0.991</u>	<u>0.997</u>	<u>0.994</u>
RPR-RHGT(2022) [23]	–	–	–	0.693	–	0.754	0.886	–	0.912	0.889	–	0.919
UPLR (2022) [12]	0.514	0.676	0.570	0.902	0.970	0.927	0.912	0.978	0.937	0.967	0.994	0.974
TG-INT	0.781	0.958	0.845	0.967	0.989	0.976	0.966	0.989	0.975	0.993	0.998	0.995

baseline. These include MTransE [14], IPTransE [15], BootEA [16], NAEA [17] and AliNet [18] which use only graph embeddings. Also include JAPE [19], GCN-Align [21], MultiKE [20], BERT-INT [11], RPR-RHGT [23] and UPLR [12] which use various side information. To facilitate comparison experiments, we chose Hit@k (k=1,5,10), MRR (Mean Reciprocal Rank), and MR (Mean Rank) as the metrics. Hit@k indicates the percentage of correctly matched entity pairs ranked in the top k, and Hit@1 is the accuracy. Both Hit@k and MRR have higher values to indicate better performance, while MR is the opposite.

The results of the TG-INT experiments on the CVD19-6K and DBP15K datasets are shown in Table 2. The number in **bold** denotes the best results of all models and second best results are underline. We conducted our experiments on CVD19-6K using OpenEA [26] and also derived the results of the BERT-INT from our experiments, using the same parameters and environment for comparison. The rest is taken from the original papers. The experimental results indicate that the model based solely on graph embeddings performs poorly on the vulnerability KG due to limited interaction between entities and relations, resulting in less semantic information compared to a general KG. However, the alignment is greatly improved by models that incorporate side information, such as text. It should be noted that while alignment based solely on graph embeddings is inadequate, they can still provide a useful reference. Our TG-INT model outperforms all baseline models, and the inclusion of graph structure information performs better than BERT-INT, which relies solely on text information.

Comparative Experiments on CVD19-6K. The vulnerability KG contains a large number of entities that are very similar to one another, and the relations connecting these entities are not very diverse. This not only makes entity alignment

methods based purely on graph structure ineffective, but also hinders traditional methods based on text matching.

We utilized Jaccard and edit distance algorithms to compute the similarity, and the results are shown in Table 3. As depicted, the character similarity-based technique outperforms the graph embedding-based approach. (char) indicates that all entities are compared based on characters, whereas name/des implies that for entities such as vulnerability numbers, their names or descriptions are employed instead of the numbers. Although the edit distance approach is simple yet effective, it is still not entirely satisfactory.

We also conducted EA experiments using graph embeddings, and evaluated them using cosine similarity. In addition, we employed the RA approach, which involves sharing relation embeddings between two graphs. Our proposed method, QuatAE, outperforms QuatE by 1.8% in terms of Hit@1. Furthermore, QuatAE achieves the best performance in CVD19-6K compared to the other alignment models in Table 2 that use only graph embeddings.

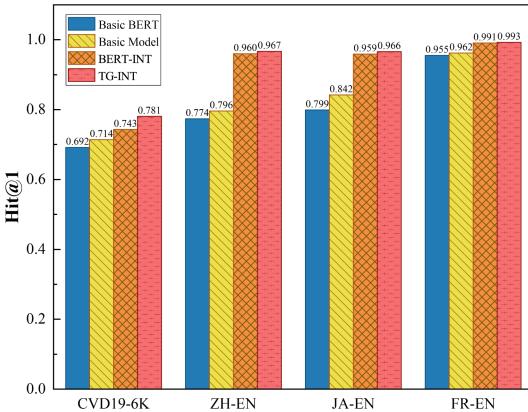
Table 3. Entity alignment based text-only and graph-only.

Model	CVD19-6K				
	Hit@1	H@5	H@10	MR	MRR
Only use text information.					
Jaccard (char)	0.260	0.349	0.383	871	0.304
Jaccard (char/name)	0.428	0.526	0.560	813	0.476
Jaccard (char/des)	0.532	0.654	0.694	450	0.590
Edit distance (char)	0.314	0.404	0.439	994	0.355
Edit distance (char/name)	0.524	0.663	0.710	418	0.587
Edit distance (char/des)	0.583	0.720	0.764	421	0.644
Only use graph embeddings.					
TransE (RA)	0.068	0.094	0.125	1298	0.090
QuatE (RA)	0.228	0.300	0.317	2491	0.263
QuatAE	0.246	0.313	0.333	1832	0.280

Ablation Study. We conducted ablation experiments on TG-INT and its basic model, and the results are presented in Table 4. The introduction of graph embeddings (GE) significantly improved the alignment of CVD19-6K and DBP15K datasets in the basic model, with an accuracy improvement of 0.7%–4.3%. After incorporating multi-view interaction, the effect of graph embeddings remained significant, resulting in a 3.8% accuracy improvement on CVD19-6K and outperforming the case without graph embeddings on the DBP15K datasets. The effect of introducing graph embeddings on CVD19-6K is illustrated in Fig. 5.

Table 4. Ablation study on CVD19-6K and DBP15K.

Model	CVD19-6K			DBP15K _{ZH_EN}			DBP15K _{JA_EN}			DBP15K _{FR_EN}		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
Basic Model (w/o GE)	0.692	0.929	0.775	0.774	0.951	0.841	0.799	0.954	0.846	0.955	0.993	0.971
Basic Model	0.714	0.950	0.798	0.796	0.959	0.856	0.842	0.971	0.891	0.962	0.996	0.979
TG-INT (w/o GE)	0.743	0.943	0.814	0.960	0.982	0.969	0.959	0.984	0.970	0.991	0.997	0.994
TG-INT	0.781	0.958	0.845	0.967	0.989	0.976	0.966	0.989	0.975	0.993	0.998	0.995

**Fig. 5.** Entity alignment with the addition of graph embeddings.

6 Conclusion

In this paper, we propose a model called TG-INT that uses a multi-view approach to align entities in vulnerability KGs by combining text-graph, neighbor, and attribute views. To achieve this, we use QuatAE, a graph embedding method, and a straightforward technique to combine graph and text embeddings. Our experiments show that adding graph embeddings to the entity alignment process provides valuable information without introducing significant noise. Our approach works not only for vulnerability KGs but also for general knowledge graphs, resulting in improved performance.

Acknowledgments. This work is funded by the National Key Research and Development Plan (Grant No. 2021YFB3101704), the National Natural Science Foundation of China (No. 62272119, 62072130, U20B2046), the Guangdong Basic and Applied Basic Research Foundation (No.2023A1515030142, 2020A15150104 50, 2021A1515012307), Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019), and Guangdong Higher Education Innovation Group (No. 2020KCXTD007), Guangzhou Higher Education Innovation Group (No. 202032854), Consulting project of Chinese Academy of Engineering (2022-JB-04-05, 2021-HYZD-8-3), the Eleventh Key Project of Education Teaching Reform in Guangzhou Municipality.

References

1. Huang, X., Zhang, J., Li, D., et al.: Knowledge graph embedding based question answering. In: WSDM, pp. 105–113 (2019)
2. Dimitriadis, I., Poiitis, M., Faloutsos, C., et al.: TG-OUT: temporal outlier patterns detection in Twitter attribute induced graphs. *World Wide Web* **25**, 2429–2453 (2022)
3. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: probabilistic alignment of relations, instances, and schema. In: VLDB, pp. 157–168 (2012)
4. Sassi, S., Tissaoui, A., Chbeir, R.: LEOnto+: a scalable ontology enrichment approach. *World Wide Web* **25**, 2347–2378 (2022)
5. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
6. Zhang, S., Tay, Y., Yao, L., et al.: Quaternion knowledge graph embeddings. In: NIPS (2019)
7. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
8. Li, R., Cao, Y., Zhu, Q., et al.: How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. In: AAAI, pp. 5781–5791 (2022)
9. Wang, H., Lian, D., Zhang, Y., et al.: Binarized graph neural network. *World Wide Web* **24**, 825–848 (2021)
10. Zhang, Z., Chen, J., Chen, X., et al.: An industry evaluation of embedding-based entity alignment. In: COLING, pp. 179–189 (2020)
11. Tang, X., Zhang, J., Chen, B., et al.: BERT-INT: a BERT-based interaction model for knowledge graph alignment. In: IJCAI (2020)
12. Li, J., Song, D.: Uncertainty-aware pseudo label refinery for entity alignment. In: Proceedings of the ACM Web Conference, pp. 829–837 (2022)
13. Liu, X., Hong, H., Wang, X., et al.: SelfKG: self-supervised entity alignment in knowledge graphs. In: Proceedings of the ACM Web Conference, pp. 860–870 (2022)
14. Chen, M., Tian, Y., Yang, M., et al.: Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In: IJCAI, pp. 1511–1517 (2017)
15. Zhu, H., Xie, R., Liu, Z., et al.: Iterative entity alignment via joint knowledge embeddings. In: IJCAI, pp. 4258–4264 (2017)
16. Sun, Z., Hu, W., Zhang, Q., et al.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI, pp. 4396–4402 (2018)
17. Zhu, Q., Zhou, X., Wu, J., et al.: Neighborhood-aware attentional representation for multilingual knowledge graphs. In: IJCAI, pp. 3231–3237 (2019)
18. Sun, Z., Wang, C., Hu, W., et al.: Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In: AAAI (2020)
19. Sun, Z., Hu, W., Li, C., et al.: Cross-lingual entity alignment via joint attribute preserving embedding. In: ISWC, pp. 628–644 (2017)
20. Zhang, Q., Sun, Z., Hu, W., et al.: Multi-view knowledge graph embedding for entity alignment. In: IJCAI, pp. 5429–5435 (2019)
21. Wang, Z., Lv, Q., Lan, X., et al.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: EMNLP, pp. 349–357 (2018)
22. Wu, Y., Liu, X., Feng, Y., et al.: Relation-aware entity alignment for heterogeneous knowledge graphs. In: IJCAI, pp. 5278–5284 (2019)

23. Cai, W., Ma, W., Zhan, J., et al.: Entity alignment with reliable path reasoning and relation-aware heterogeneous graph transformer. In: IJCAI, pp. 1930–1937 (2022)
24. Huang, H., Li, C., Peng, X., et al.: Cross-knowledge-graph entity alignment via relation prediction. *Knowl. Based Syst.* **240**, 107813 (2022)
25. Xiong, C., Dai, Z., Callan, J., et al.: End-to-end neural ad-hoc ranking with kernel pooling. In: SIGIR, pp. 55–64 (2017)
26. Sun, Z., Zhang, Q., Hu, W., et al.: A benchmarking study of embedding-based entity alignment for knowledge graphs. In: Proceedings of the VLDB Endowment, pp. 2326–2340 (2020)