Graph Hierarchy and Language Model-Based Explainable Entity Alignment of Knowledge Graphs

Kunyoung Kim[a], Donggyu Kim[b] and Mye Sohn[c,*]
Department of Industrial Engineering, Sungkyunkwan University, Suwon, Korea

Keywords: Entity Alignment, Explainable AI, Knowledge Graph, Language Model.

Abstract: This paper proposes a Graph hierarchy and Language model-based Explainable Entity Alignment (GLEE) framework to perform Entity Alignment (EA) between two or more Knowledge Graphs (KGs) required for solving complex problems. Unlike existing EA methods that generate embedding for entities using KG structure information to calculate the similarity between entities, the GLEE framework additionally utilizes graph hierarchy and datatype properties to find entities to be aligned. In the GLEE framework, the semantically similar hyper-entities of the entities to be aligned are discovered to reflect graph hierarchy in the alignment. Also, the semantically similar datatype properties and their values of the entities are also utilized in EA. At this time, language model is utilized to calculate semantic similarity of the hyper-entities or datatype properties. As a result, the GLEE framework can trustworthy explain why the two entities are aligned using the subgraphs that consist of similar hyper-entities, semantically identical properties, and their data values. To show the superiority of the GLEE framework, the experiment is performed using real world dataset to prove the EA performance and explainability.

1 INTRODUCTION

As knowledge graphs (KGs) are proven to help integrate and access disparate data sources using machine learning and inference capabilities, various KGs have been developed, ranging from KGs for research to enterprise KGs (Zou, 2020). However, due to the complexity of the problems to be solved using KGs, it is generally difficult for a single KG to meet the diverse knowledge requirements of an application (Zeng et al., 2021). To leverage different KGs simultaneously, a research, Entity Alignment (EA), has been conducted to identify and connect entities that belong to different KGs but refer to the same real-world concept (Fanourakis et al., 2023). Traditionally, EA has been performed using network and iteration-based approaches, but recently, Knowledge Graph Representation Learning (KGRL), which generates embeddings of entities based on graph structures and calculates the similarity between entities using the embeddings, has become a representative player (Zeng et al., 2021). However, KGRL, which is performed based on the structural similarity of KGs, may encounter the following difficulties.

First, to solve complex problems in which multiple elements are intertwined, EA between KGs with various domains and scopes must be performed. At this point, it is difficult to assume that all KGs have a similar structure (Wang et al., 2022). Therefore, a method that can perform EA regardless of the structure or scope of KGs is needed.

Second, as a side-effect, it is not easy to explain the results of EA on two KGs with different structures or domains. Since different KGs often have different naming schemes, it is hard to look up the properties and neighbors of the aligned entities to check whether the alignment is correct or not (Trisedya et al., 2023). To help checking the EA results, explanation is needed.

To do so, this paper proposes a novel Graph hierarchy and Language model-based Explainable

[a] https://orcid.org/0000-0002-3570-8645
[b] https://orcid.org/0009-0002-5753-5084
[c] https://orcid.org/0000-0002-1951-3493
* Corresponding author

Kim, K., Kim, D. and Sohn, M.
Graph Hierarchy and Language Model-Based Explainable Entity Alignment of Knowledge Graphs.
DOI: 10.5220/0013012300003886
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 1st International Conference on Explainable AI for Neural and Symbolic Methods (EXPLAINS 2024)*, pages 124-131
ISBN: 978-989-758-720-7
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

Entity alignment (GLEE) framework. Even though graph hierarchies (Zhang et al., 2020) and datatypes (Kim et al., 2022; Shen et al., 2021) contain crucial information about entities, this information is not directly reflected in KGRL EA methods. In the GLEE framework, they are used to improve the EA performance and provide explanation. The GLEE framework consists of three steps.

In the first step, after selecting two KGs to be aligned, candidate entities that can be aligned with a specific entity in one KG (hereafter, target entity) are found in the other KG. To do so, this paper calculates graph structural similarity (hereafter, S-similarity) using the existing embedding-based EA method. S-similarity is not performed for EA directly, but for finding candidate entities.

In the second step, EA is performed after finding the entity to be aligned with the target entity among candidate entities. To do so, this paper devises graph hierarchical similarity (H-similarity) and datatype similarity (D-similarity). H-similarity is calculated to discover common hyper-concepts shared by the target entity and the candidate entity. It has a larger value when the same hyper-entities appear in fewer hops. At this time, BERT (Devlin et al., 2018)-based word embedding is used on the entity names to determine whether the two hyper-entities are the same. If two KGs are from different languages, multilingual BERT model (Pires et al., 2019) is used. D-similarity is calculated by discovering common properties of the entities and counting the set of common properties with the same datatype values. To determine whether the two properties are same, BERT-based word embedding is used on the property names, too. The second step results are similarities between the target entity and candidate entities.

In the third step, the EA is performed using the three kinds of similarities and explanation is performed using two subgraphs of the target entity and its aligned entity extracted from each of the two KGs. The subgraphs are extracted from the different KGs but have same graph structure and contents. In GLEE framework, the subgraphs consist of the paths with the aligned entities as the starting node and the hyper concept common to both subgraphs as the ending node, and the datatype properties and their data values of the aligned entities. The subgraphs are used to explain the EA results.

The main contribution of this paper is providing a detailed steps to utilize three kinds of similarity to improve the EA performance and provide explanation. Since S-similarity has relatively low computational cost than H-similarity and D-similarity to find the similar entities in the large KGs, S-similarity is utilized in advance to discover the candidate pairs of entities to be aligned. On the other hand, H-similarity and D-similarity are more accurate and interpretable than S-similarity, but it is impossible to directly apply them to align entities in the large KGs. Therefore, they are applied in the candidate pairs obtained by using S-similarity.

This paper is organized as follow. The related works of this paper is presented in Section 2. In Section 3 the illustrative scenario is provided to show how GLEE framework works. Section 4 depicts GLEE framework with the details. The superiority of the framework is presented in Section 5 with experiments. Finally, Section 6 proposes the conclusions and further research.

2 RELATED WORKS

Many studies have been performed to automatically align entities of different knowledge graphs (KGs). TransE (Bordes et al., 2013) treats the relation in a relationship triple as the translation from head to the tail entities achieving promising results in one-to-one relation. However, it cannot consider the multi-hop relationships nor process the complex relationships such as one-to-many, many-to-one, many-to-many. Therefore, many variants of TransE have subsequently appeared, such as TransR (Lin et al., 2015), TransH (Wang et al., 2014), TransD (Ji et al., 2015), TransG (Xiao et al., 2015a), TransA (Xiao et al., 2015b), and PTransE (Li et al., 2020). These studies tackle different limitations of TransE and enhance the ability to model structures within KGs.

Recently, due to the growth of the deep learning methods, graph convolutional network (GCN) (Kipf & Welling, 2016)-based methods are widely adopted for entity alignment. GCN-Align (Wang et al., 2018) captures the entities' neighborhood structures for the first time by utilizing GCNs and achieving promising results. RN-GCN (Wu et al., 2019a) incorporates relation information via attentive interactions between the KG and its dual relation counterpart. HGCN (Wu et al., 2019b) jointly learns entity and relation representations without requiring pre-aligned relations. EMGCN (Tam et al., 2020) perform unsupervised EA by using both relation and attribute information.

Several recent studies use semi-supervised learning that generates new pre-aligned seeds during the iteration process. MCEA (Qi et al., 2023) uses a multiscale graph convolution model to embed the graph and uses intermediate results to guide the negative sampling process. PEEA (Tang et al., 2023) increases the connections between far-away entities and labeled ones by incorporating positional information into the representation learning.

Many studies have been performed to improve the EA performance. These studies assume that the KGs have similar structure and perform experiment using similar KGs (i.e., DBpedia English and DBpedia French). That is, the performance is not guaranteed when the KGs have different structure and scope. Also, they do not provide explanation of EA results which undermines the reliability.

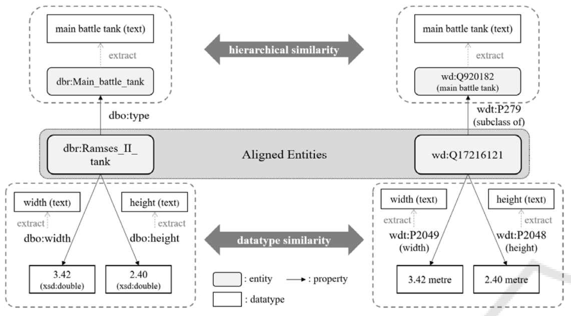

Figure 1: An illustrative example of the GLEE framework.

3 AN ILLUSTRATIVE EXAMPLE

An illustrative example is used to show how the GLEE framework conducts EA and how to explain its results. As depicted in Figure 1, two entities representing the same real-world concept, the 'Ramses II tank,' a specific model of a main battle tank, are selected from different KGs. The entity 'dbr:Ramses_II_tank' is from DBpedia and the entity 'wd:Q17216121' is from Wikidata.

First, the entities that represent the upper concepts of 'Ramses II tank' are identified to determine their similarity in the graph hierarchy. At this time, specific properties that represent relation to upper concepts (i.e., 'dbo:type' and 'rdf:type' in DBpedia) are used. In the given example, the entities have 'main battle tank' as a common hyper concept. If the same hyper-concept does not exist in 1-hop relations for the entities, entities of 2 or more hops are discovered and compared. Second, to determine their similarity depending on their datatypes, the common properties for the two entities are discovered. In the example, the two entities have common properties, 'width' and 'height.' In addition, the datatype values for the properties are also same. Therefore, we can determine that two entities in different KGs represent the same real-world concept because they have a common hyper-concept and datatype. Furthermore, the explanation using this result is also trustworthy.

4 GLEE FRAMEWORK

4.1 Overall Framework

The overall systematic procedure of the GLEE framework is depicted in Figure 2. The GLEE framework consists of three modules: Graph Structure-based Candidate Entity Discovery Module, Entity Similarity Assessment Module, and Subgraph-based Explanation Module.

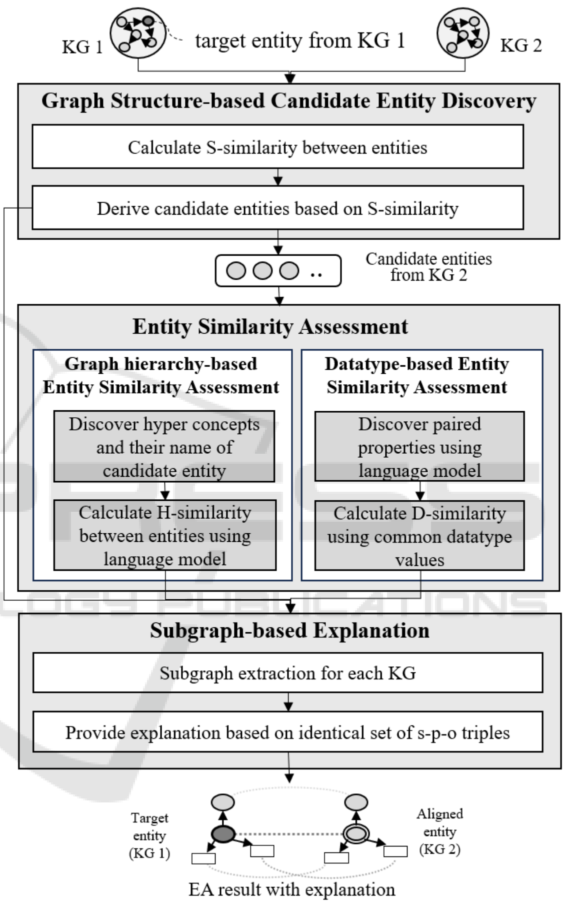
KG 1
target entity from KG 1
KG 2
Graph Structure-based Candidate Entity Discovery
Calculate S-similarity between entities
Derive candidate entities based on S-similarity
Candidate entities from KG 2
Entity Similarity Assessment
Graph hierarchy-based Entity Similarity Assessment
Discover hyper concepts and their name of candidate entity
Calculate H-similarity between entities using language model
Datatype-based Entity Similarity Assessment
Discover paired properties using language model
Calculate D-similarity using common datatype values
Subgraph-based Explanation
Subgraph extraction for each KG
Provide explanation based on identical set of s-p-o triples
Target entity (KG 1)
Aligned entity (KG 2)
EA result with explanation

Figure 2: Systematic procedure of the GLEE framework.

4.2 Graph Structure-Based Candidate Entity Discovery

In this module, candidate entities that can be aligned with the target entity on $KG_1$are found from KG_2. To do so, the S-similarity between the target entity on KG_1 and all entities on KG_2 is calculated using the existing embedding-based EA methods.

The target entity ($e_{KG_1}^{(t)}$) can be simply represented as follows.

$$(e_{KG_1}^{(t)}) \in E_{KG_1} \tag{1}$$

where E_{KG_1} is the set of entities of KG_1. At this time, $e_{KG_1}^{(t)}$ is an arbitrary entity on KG_1.

For ($e_{KG_1}^{(t)}$), the S-similarity with $e_{KG_2(i)}$ (for $\forall i, i \in \mathbb{N}$) is calculated using cosine similarity as follows.

$$Ssim\left(e_{KG_1}^{(t)}, e_{KG_2(i)}\right) = \frac{GE_1\left(e_{KG_1}^{(t)}\right) \cdot GE_2\left(e_{KG_2(i)}\right)}{\left\|GE_1\left(e_{KG_1}^{(t)}\right)\right\| \left\|GE_2\left(e_{KG_2(i)}\right)\right\|} \tag{2}$$

where $e_{KG_2(i)}$ represents the i^{th} entity in KG_2 ($e_{KG_2(i)} \in E_2$, where E_2 is the set of entities of KG_2). $GE_1\left(e_{KG_1}^{(t)}\right)$ and $GE_2\left(e_{KG_2(i)}\right)$ represent the embeddings of $e_{KG_1}^{(t)}$ on KG_1 and $e_{KG_2(i)}$ in KG_2, which are generated by the existing EA method. $\left\|GE_1\left(e_{KG_1}^{(t)}\right)\right\|$ and $\left\|GE_2\left(e_{KG_2(i)}\right)\right\|$ represent the lengths of the vectors $GE_1\left(e_{KG_1}^{(t)}\right)$ and $GE_2\left(e_{KG_2(i)}\right)$.

Based on S-similarity, this paper designates the top- k entities with high S-similarity values as candidate entities for $e_{KG_1}^{(t)}$. The algorithm discovering the candidate entities for $e_{KG_1}^{(t)}$ is as follows.

Algorithm 1.

Input: KG_1, KG_2, and the target entity ($e_{KG_1}^{(t)}$)
Output: a candidate set of top-k entities on KG_2 that have high S-similarity values with $e_{KG_1}^{(t)}$

Step 1 Embedding generation. Generate embeddings of the entities using a KGRL model
Step 2 S-similarity calculation. Calculate the cosine similarity between the embeddings of the $e_{KG_1}^{(t)}$ and each entity on KG_2 ($e_{KG_2,i}$)
Step 3: Top-k selection. Identify the candidate set that is composed of the entities in KG_2 that has high similarity with $e_{KG_1}^{(t)}$

4.3 Entity Similarity Assessment

This module comprises two parts: graph hierarchy-based and datatype-based entity assessment.

4.3.1 Graph Hierarchy-Based Entity Similarity Assessment

This submodule assesses whether the candidate entities belong to the same or similar class with $e_{KG_1}^{(t)}$. To do so, graph hierarchy-based similarity (H-similarity) is calculated between $e_{KG_1}^{(t)}$ and entities in the candidate set. At this time, H-similarity is calculated by gradually increasing the number of hops of the link connected to $e_{KG_1}^{(t)}$. First, 1-hop hyper entities of $e_{KG_1}^{(t)}$ ($UE^1\left(e_{KG_1}^{(t)}\right)$) discovers as follows.

$$UE^1\left(e_{KG_1}^{(t)}\right) = \{ue_m^1 |\, pr_m^1 \in UP_1\},\ m \in \mathbb{N} \tag{3}$$

where $< e_{KG_1}^{(t)}, pr_m^1, ue_m^1 >$ is m^{th} subject-property-object (s-p-o) triple of KG_1. ue_m^1 and pr_m^1 are m^{th} hyper entity of 1-hop and its property. UP_1 is the set of properties of KG_1 that depict hyper-hypo relations. In DBpedia, a representative KG, 'dbo:type' or 'rdf:type' are examples of these properties.

Like above, 1-hop hyper entities of $e_{KG_2(i')}$ ($UE^1\left(e_{KG_2(i')}\right)$) discovers as follows.

$$\left(UE^1\left(e_{KG_2(i')}\right)\right) = \{ue_n^1 |\, pr_n^1 \in UP_2\}, n \in \mathbb{N} \tag{4}$$

where $< e_{KG_2(i')}, pr_n^1, ue_n^1 >$ is n^{th} s-p-o of KG_2. ue_n^1 and pr_n^1 are n^{th} hyper entity of 1-hop and its property. UP_2 is the set of properties of KG_2 that depict hyper-hypo relations ($i' \in \mathbb{N}$).

In the GLEE framework, it is assumed that the UP_1 and UP_2 are already discovered. Applying the same method to the 1-hop hyper entities, hyper entities with 2 or more hops can also be discovered as follows: $UE^p\left(e_{KG_1}^{(t)}\right)$ and $UE^q\left(e_{KG_2(i')}\right)$ ($p = 1,..,P, q = 1,..,Q$).

To compare hyper entities of $e_{KG_1}^{(t)}$ and $e_{KG_2(i')}$, entity name-based BERT embeddings are generated. Since entity names of hyper concepts (e.g. 'ship' and 'tank') are typically abstract compared to entity names of hypo concepts (e.g. 'USS Inchon' and 'Ramses 2 tank'), it can be assumed that BERT works well on names of hyper entities.

As a next, H-similarity between $e_{KG_1}^{(t)}$ and $e_{KG_2(i')}$ ($Hsim\left(e_{KG_1}^{(t)}, e_{KG_2(i')}\right)$) is calculated based on the number of hops and BERT-based word embeddings as follows.

$$Hsim\left(e_{KG_1}^{(t)}, e_{KG_2(i')}\right) = \max_{m,n,p,q} \frac{sim\left(ue_m^p, ue_n^q\right)}{M(p,q)} \tag{5}$$

where ue_m^p is the m^{th} entity of p hops in $UE^p\left(e_{KG_1}^{(t)}\right)$ and ue_n^q is the n^{th} entity of q hops in

$UE^q\left(e_{KG_2(i')}\right)$. $sim(ue_m^p, ue_n^q)$ is the BERT embedding-based cosine similarity between the names of the ue_m^p and ue_n^q. $M(p,q)$ represents the maximum function as follows.

$$M(p,q) = \begin{cases} p\ (p \geq q) \\ q\ (p < q) \end{cases} \tag{6}$$

At this time, the two hyper entities from KG_1 and KG_2 that maximize the score ($\underset{ue_m^p, ue_n^q}{\operatorname{argmax}} \frac{sim(ue_m^p, ue_n^q)}{M(p,q)}$), in other words, the hyper entities used to calculate the H-similarity, are defined as the common hyper entities in KG_1 and KG_2 and will be utilized in explanation.

4.3.2 Datatype-Based Entity Similarity Assessment

In this sub-module, D-similaity is calculated to compare the object values of $e_{KG_1}^{(t)}$ with those of $e_{KG_2(i')}$ $(for\ \forall\ i')$. At this time, the objects of $e_{KG_1}^{(t)}$ and $e_{KG_2(i')}$ are limited to those with a 'datatype property.' In other words, the objects are data values, such as integers, characters, or strings. For all s-p-o triples of KG_1 and KG_2, a set of properties with data values as objects is found for each as follows.

$$DP_1\left(e_{KG_1}^{(t)}\right) = \{pr_m |\ d_m \in D\} \tag{7}$$

$$DP_2\left(e_{KG_2(i')}\right) = \{pr_n |\ d_n \in D\} \tag{8}$$

where d_m and d_n are m^{th} and n^{th} data values of the objects of s-p-o triples of KG_1 and KG_2. D is a set of data values.

For each pr_m (for $\forall\ m$), this paper finds a property that is semantically similar to pr_m among the elements of $DP_2\left(e_{KG_2(i')}\right)$. To do so, BERT-based similarity between the properties is calculated. As a result, this paper obtains the semantically similar property to pr_m $(ssp(pr_m))$ in $DP_2\left(e_{KG_2(i')}\right)$:

$$ssp(pr_m) = \begin{cases} NULL,\ if\ \underset{pr_n}{\operatorname{argmax}}\, sim(pr_m, pr_n) < \delta \\ \underset{pr_n}{\operatorname{argmax}}\, sim(pr_m, pr_n), \qquad o/w \end{cases} \tag{9}$$

where $sim(pr_m, pr_n)$ is the cosine similarity between BERT embeddings of the names of pr_m and pr_n. At this time, if the $\underset{pr_n}{\operatorname{argmax}}\, sim(pr_m, pr_n)$ value is smaller than predefined threshold δ , pr_n is determined to be not the same as pr_m.

Next, the data value of $ssp(pr_m)$ and that of pr_m are extracted and compared. If the data value is a string of natural language texts, BERT embedding is utilized to determine whether they are same or not. As a result, this paper identifies data value pairs consisting of a data value of pr_m and that of $ssp(pr_m)$ for all m. At this time, the two elements of every pair must be identical or semantically similar. After counting the number of elements of $DP_1\left(e_{KG_1}^{(t)}\right)$ and the identified data pairs, the D-similarity is calculated as follows.

$$Dsim = \frac{no(\text{data pairs with same data values})}{no(\text{elements of } DP_1\left(e_{KG_1}^{(t)}\right))} \tag{10}$$

The D-similarity calculation procedure is summarized as following algorithm. candidate entities for $e_{KG_1}^{(t)}$ is as follows.

Algorithm 2.

Input: sets of the properties of $e_{KG_1}^{(t)}$ and $e_{KG_2(i')}$ that have data values as objects
Output: D-similarity between two object values.

Step 1 Property finding For all s-p-o triples of KG_1 and KG_2, find a set of properties with data values as objects and set them to $DP_1\left(e_{KG_1}^{(t)}\right)$ and $DP_2\left(e_{KG_2(i')}\right)$, respectively
Step 2 Property matching For each pr_m, find semantically identical property of it among the properties of $DP_2\left(e_{KG_2(i')}\right)$ and set it to $ssp(pr_m)$ (for $\forall\ m$)
Step 3 Find data value of matched properties Extract data value linked to pr_m and $ssp(pr_m)$ and identify whether they are same or not
Step 4: D-similarity Calculation Calculate the D-similarity based on the ratio of the matched data values to the total number of datatype properties of $e_{KG_1}^{(t)}$

4.4 Subgraph-Based Explanation

Using the S-similarity, H-similarity, and D-similarity, one of the candidate entities of KG_2 is determined as the aligned entity of $e_{KG_1}^{(t)}$. The aligned entity of $e_{KG_1}^{(t)}$ ($AE\left(e_{KG_1}^{(t)}\right)$)is determined as follows.

$$AE\left(e_{KG_1}^{(t)}\right) = \underset{e_{KG_2(i')}}{\operatorname{argmax}}\, \alpha \cdot Ssim\left(e_{KG_1}^{(t)}, e_{KG_2(i')}\right) + \beta \cdot Hsim\left(e_{KG_1}^{(t)}, e_{KG_2(i')}\right) + \gamma \cdot Dsim\left(e_{KG_1}^{(t)}, e_{KG_2(i')}\right) \quad (11)$$

where α, β, and γ are the hyperparameters ($0 \leq \alpha, \beta, \gamma \leq 1, \alpha + \beta + \gamma = 1$).

To explain the EA results, this paper utilizes the subgraphs of the target entity and the corresponding aligned entity extracted from KG_1 and KG_2, respectively. The subgraphs are generated with two phases as follows. First, this paper generates paths from for the target and aligned entity to the upper entities they have in common, respectively. Second, for the target and aligned entity, the paths are expanded using the semantically similar datatype properties and their data values, respectively. An example of the subgraphs is illustrated in Figure 1

As shown in Figure 1, the two subgraphs are extracted from different KGs but are composed of semantically similar properties and entities. The upper entities, properties, and data values (a.k.a. s-p-o) that are common to the two subgraphs, are used to explain why two entities in different KGs are aligned. For example, in Figure 1, 'dbr:Ramses_II_tank' in KG_1 and 'wd:Q17216121' in KG_2 are aligned because they have the hyper entity with the same entity name (main battle tank) and the common datatype properties with the same data values (width 3.42 and height 2.40).

5 PERFORMANCE EVALUAITON

To demonstrate the GLEE framework's performance, this paper selects entities of DBpedia and Wikidata. DBpedia and Wikidata are KGs that are generated based on Wikipedia but have different term usage and structure since they are developed and maintained by different organizations. The selected entities belong to 5 categories: 'vessels,' 'tanks,' 'aircrafts' 'generals,' and 'military units.'

For the entities, we generate the pairs of entities that one belongs to DBpedia and the other belongs to Wikidata. Some pairs may consist of entities depicting the same real-world concept ('exactly same' in the table), while others may not. Depending on the entities, entity pairs are classified into three types: exactly same pair, intra-category pair, and inter-category pair. The first is the pairs consist of two entities which are semantically same, the second are pairs consist of entities that are not same and belong to the same category, and the third are pairs consist of entities that are not same and belong to different categories.

For the three kinds of entity pairs, this paper calculates H-similarity and D-similarity. For each category, this paper calculates the average values of H-similarity and D-similarity for the three types of pairs. At this time, D-similarity is calculated for DBpedia and Wikidata respectively, since D-similarity depends on the number of datatype properties of the target entity. One regards the entity of DBpedia in the pair as the target entity, and the other regards the entity of Wikidata in the pair as the target entity. This paper does not calculate S-similarity in the experiment since it is just applying existing KGRL methods, and we do not have contribution to it. The similarities are summarized in Table 1.

Table 1: Similarity of the pairwise entities.

Category		$Hsim$	$Dsim$ (DBpedia)	$Dsim$ (Wikidata)
Vessels	Exactly same	**0.795**	**0.733**	**0.733**
	Intra-category	0.527	0.000	0.000
	Inter-category	0.505	0.000	0.000
Tanks	Exactly same	**0.972**	**0.403**	**0.594**
	Intra-category	0.783	0.000	0.000
	Inter-category	0.647	0.000	0.000
Aircrafts	Exactly same	**0.753**	**0.345**	**0.424**
	Intra-category	0.629	0.100	0.143
	Inter-category	0.613	0.000	0.000
Generals	Exactly same	**0.683**	**0.793**	**0.442**
	Intra-category	0.661	0.000	0.000
	Inter-category	0.479	0.000	0.000
Units	Exactly same	**0.917**	**0.572**	**0.642**
	Intra-category	0.734	0.000	0.000
	Inter-category	0.546	0.000	0.000

For all entities, 'exactly same' entity pairs have the highest H-similarity and D-similarity values. It is proven that the proposed similarities are effective for EA. In case of H-similarity, 'intra-category' has higher H-similarity value than that of 'inter-category.' It is because intra-category pairs have relatively short paths to the common hyper entity than inter-category pairs. In case of D-similarity, values of 'intra-category' and 'inter-category' are almost zero. That is, it is hard to have the same datatype property with same data value of the entities that depict different real-world concepts.

The core contribution of the GLEE framework is to provide explanation of EA results using graph hierarchy and datatypes. To show how GLEE framework explains the EA result, this paper selects a pair of entities as an example that was aligned in the experiment. The entities depict 'Leopard 2' which is a tank and extracted the subgraphs from DBpedia and Wikidata. The subgraphs are depicted in Figure 3. Even though the two graphs are extracted from the different KGs, they both have the hyper entities which depict 'main battle tank.' Also, the datatype properties in the subgraphs have the same meaning and the datatype values are same. Using these subgraphs, it can be acknowledged that the aligned entities depict the same real-world concept.

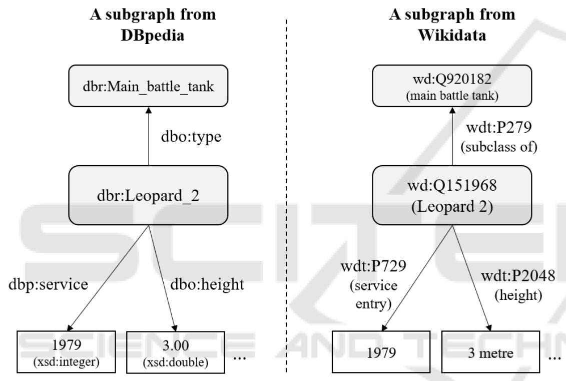

Figure 3: An example of generated subgraphs.

6 CONCLUSIONS

This paper proposed a method to align the entities of different KGs not only using the graph structure, but also using the graph hierarchy and datatypes. In the proposed method, graph structure-based EA model is applied to derive the candidate pairs of entities to be aligned, and the information on graph hierarchy and datatypes is utilized to find the exact pairs. By doing so, alignment performance can be improved, and explanation of the alignment can also be provided. However, this paper has the following limitations. First, it cannot provide an explanation in perspective of graph structural similarity. That is, the information on the relations with multi-hop neighbor entities is not provided as an explanation. Second, the impact of each property, object or datatype to the alignment is not provided, whether they are critical for alignment or somewhat meaningless. Therefore, in our further research, we will develop a method to discover neighbor entities which are crucial for alignment and provide weights that depict importance of properties and objects of the aligned entities.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00275408).

REFERENCES

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

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fanourakis, N., Efthymiou, V., Kotzinos, D., & Christophides, V. (2023). Knowledge graph embedding methods for entity alignment: experimental review. *Data Mining and Knowledge Discovery*, 37(5), 2070-2137.

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015, July). Knowledge graph embedding via dynamic mapping matrix. In *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)* (pp. 687-696).

Kim, J., Kim, K., Sohn, M., & Park, G. (2022). Deep Model-Based Security-Aware Entity Alignment Method for Edge-Specific Knowledge Graphs. *Sustainability*, 14(14), 8877.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Li, Y., Qin, D., & Yang, X. (2020, December). Path modeling based on entity-connectivity for knowledge base completion. *In 2020 7th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 984-989). IEEE.

Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., & Liu, S. (2015). Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*.

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT?. *arXiv preprint arXiv:1906.01502*.

Qi, D., Chen, S., Sun, X., Luan, R., & Tong, D. (2023). A multiscale convolutional gragh network using only structural information for entity alignment. *Applied Intelligence*, *53*(7), 7455-7465.

Shen, Y., Li, Z., Wang, X., Li, J., & Zhang, X. (2021, October). Datatype-aware knowledge graph representation learning in hyperbolic space. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 1630-1639).

Tam, N. T., Trung, H. T., Yin, H., Van Vinh, T., Sakong, D., Zheng, B., & Hung, N. Q. V. (2020). Entity alignment for knowledge graphs with multi-order convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(9), 4201-4214.

Tang, W., Su, F., Sun, H., Qi, Q., Wang, J., Tao, S., & Yang, H. (2023, February). Weakly supervised entity alignment with positional inspiration. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (pp. 814-822).

Trisedya, B. D., Salim, F. D., Chan, J., Spina, D., Scholer, F., & Sanderson, M. (2023). i-Align: an interpretable knowledge graph alignment model. *Data Mining and Knowledge Discovery*, 37(6), 2494-2516.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014, June). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 28, No. 1).

Wang, Z., Lv, Q., Lan, X., & Zhang, Y. (2018). Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 349-357).

Wang, S., Wei, X., Dos Santos, C. N., Wang, Z., Nallapati, R., Arnold, A., & Philip, S. Y. (2021, December). Knowledge graph representation via hierarchical hyperbolic neural graph embedding. In *2021 IEEE International Conference on Big Data (Big Data)* (pp. 540-549). IEEE.

Wang, H., Wang, Y., Li, J., & Luo, T. (2022). Degree aware based adversarial graph convolutional networks for entity alignment in heterogeneous knowledge graph. *Neurocomputing*, 487, 99-109.

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., & Zhao, D. (2019a). Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*.

Wu, Y., Liu, X., Feng, Y., Wang, Z., & Zhao, D. (2019b). Jointly learning entity and relation representations for entity alignment. *arXiv preprint arXiv:1909.09317*.

Xiao, H., Huang, M., Hao, Y., & Zhu, X. (2015a). Transg: A generative mixture model for knowledge graph embedding. *arXiv preprint arXiv:1509.05488*.

Xiao, H., Huang, M., Hao, Y., & Zhu, X. (2015b). TransA: An adaptive approach for knowledge graph embedding. *arXiv preprint arXiv:1509.05490*.

Zeng, K., Li, C., Hou, L., Li, J., & Feng, L. (2021). A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, 2, 1-13.

Zhang, Z., Cai, J., Zhang, Y., & Wang, J. (2020, April). Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 03, pp. 3065-3072).

Zou, X. (2020, March). A survey on application of knowledge graph. *In Journal of Physics: Conference Series* (Vol. 1487, No. 1, p. 012016). IOP Publishing.