

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

# RDF-star2Vec: RDF-star Graph Embeddings for Data Mining

**SHUSAKU EGAMI<sup>1</sup>, TAKANORI UGAI<sup>1,2</sup>, MASATERU OOTA<sup>1</sup>, KYOUMOTO MATSUSHITA<sup>2</sup>, TAKAHIRO KAWAMURA<sup>3</sup>, KOJI KOZAKI<sup>4</sup>, KEN FUKUDA<sup>1</sup>**

<sup>1</sup>National Institute of Advanced Industrial Science and Technology, Tokyo, Japan (e-mail: {s-egami,ken.fukuda}@aist.go.jp)

<sup>2</sup>Fujitsu Ltd., Kanagawa, Japan (e-mail: ugai@fujitsu.com)

<sup>3</sup>National Agriculture and Food Research Organization, Tsukuba, Japan (e-mail: takahiro.kawamura@naro.go.jp)

<sup>4</sup>Osaka Electro-Communication University, Osaka, Japan (e-mail: kozaki@osakac.ac.jp)

Corresponding author: Ken Fukuda (e-mail: ken.fukuda@aist.go.jp).

This paper is based on results obtained from projects, JPNP20006 and JPNP180013 commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number JP19H04168.

**ABSTRACT** Knowledge Graphs (KGs) such as Resource Description Framework (RDF) data represent relationships between various entities through the structure of triples (*<subject, predicate, object>*). Knowledge graph embedding (KGE) is crucial in machine learning applications, specifically in node classification and link prediction tasks. KGE remains a vital research topic within the semantic web community. RDF-star introduces the concept of a *quoted triple* (QT), a specific form of triple employed either as the subject or object within another triple. Moreover, RDF-star permits a QT to act as compositional entities within another QT, thereby enabling the representation of recursive, hyper-relational KGs with nested structures. However, existing KGE models fail to adequately learn the semantics of QTs and entities, primarily because they do not account for RDF-star graphs containing multi-leveled nested QTs and QT-QT relationships. This study introduces RDF-star2Vec, a novel KGE model specifically designed for RDF-star graphs. RDF-star2Vec introduces graph walk techniques that enable probabilistic transitions between a QT and its compositional entities. Feature vectors for QTs, entities, and relations are derived from generated sequences through the structured skip-gram model. Additionally, we provide a dataset and a benchmarking framework for data mining tasks focused on complex RDF-star graphs. Evaluative experiments demonstrated that RDF-star2Vec yielded superior performance compared to recent extensions of RDF2Vec in various tasks including classification, clustering, entity relatedness, and QT similarity.

**INDEX TERMS** Knowledge graph embedding; RDF2Vec; RDF-star; Hyper-relational knowledge graphs; N-ary relation; Graph walk

## I. INTRODUCTION

Knowledge graphs (KGs) such as Resource Description Framework (RDF) data represent relationships between various entities in terms of triples (*<subject, predicate, object>*), facilitating advanced search and reasoning based on semantic relationships. Despite these capabilities, RDF faces the challenge of adequately representing relations beyond the binary. To address this limitation, RDF-star (formerly spelled RDF\*) [1] has garnered considerable interest. This extension introduces the *quoted triple* (QT), a specific type of triple that can serve as the subject or object within another triple. To date, RDF-star has been implemented across

numerous triplestores<sup>1</sup>. In addition, several use cases<sup>2</sup> [2], [3] and support tools [4]–[7] for RDF-star have already been presented. Furthermore, the RDF-star Working Group<sup>3</sup> was established in August 2022 and is working on recommendations for extending both RDF and SPARQL Protocol and RDF Query Language (SPARQL)<sup>4</sup>. Therefore, it is expected that various hyper-relational KGs will be published in RDF-star format in the near future.

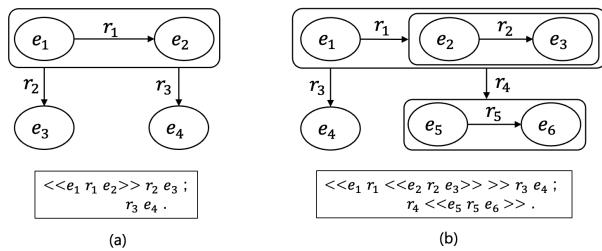
Various methods for knowledge graph embedding (KGE) have been proposed in recent years, targeting data mining

<sup>1</sup><https://w3c.github.io/rdf-star/implementations.html>

<sup>2</sup><https://w3c.github.io/rdf-star/UCR/rdf-star-ucr.html>

<sup>3</sup><https://www.w3.org/2022/08/rdf-star-wg-charter/>

<sup>4</sup>SPARQL is a recursive acronym.



**FIGURE 1.** RDF-star graphs: (a) simple RDF-star and (b) complex RDF-star graphs

applications. However, two distinct limitations have been identified in existing KGE methods.

- (1) **Inability directly handle RDF-star data:** KGE methods for RDF exist [8], [9], but methods specifically tailored for RDF-star data are conspicuously lacking. Although there are approaches to convert RDF-star triples into regular RDF [10] triples, as well as to load non-RDF data formats (e.g., CSV) [11], these techniques fail to capture the vector representations corresponding to QTs, and consequently, compromise the original semantic representation.
- (2) **Inability to handle complex RDF-star data:** RDF-star permits the representation of recursive hyper-relational KGs featuring nested structures. This is possible because QT can function as the compositional entities of another QT. However, existing hyper-relational KGE methods [11]–[13] can handle only rudimentary structures, as shown in Figure 1(a), and fall short of capturing the semantics of complex structures, including nested QT structures and interrelations between QTs, as shown in Figure 1(b). Additionally, the absence of a publicly available complex RDF-star dataset hinders the further development of RDF-star KGE methods.

This paper introduces RDF-star2Vec, a novel graph walk-based KGE method designed to learn vector representations of normal entities, relations, and QTs in RDF-star graphs with complex structures, including multi-leveled nested QTs and relations between QTs. Specifically, graph walk methods generate sequences that permit probabilistic transition between QTs and asserted triples in RDF-star graphs. Subsequently, feature vectors for QTs, entities, and relations are derived from the generated sequences employing the structured skip-gram model [14]. This approach allows for the direct embedding of RDF-star data into low-dimensional vector space, preserving the semantic representation of the original data. Importantly, the technique situates QTs and their compositional entities in close proximity within the generated sequences, thereby learning the N-ary relations represented by RDF-star. The method was realized through a Java implementation of RDF2Vec [8], [9] and has been released as open-source software<sup>5</sup>.

<sup>5</sup><https://github.com/aistairc/RDF-star2Vec>

Furthermore, we introduce a complex RDF-star dataset (KGRC-RDF-star), predicated on KGRC-RDF [15], [16], a scene KG designed for Explainable Artificial Intelligence (XAI) benchmarking and constructed from text data in mystery novels. The dataset features nested statements and scenes such as “Person A said ‘Person B saw Person C was in D’.” Moreover, four gold standard datasets have been constructed to assess classification, clustering, entity relatedness, and QT similarity tasks using the embeddings of KGRC-RDF-star and provided through GEval [17], a KGE evaluation framework. Therefore, this study will augment the existing body of knowledge concerning KGE methods for RDF-star graphs.

The remainder of this paper is structured as follows: Section II introduces related works focusing on KGE methods for RDF, methods for hyper-relational KGs, and benchmarking datasets, and outlines the limitations of these prior works and the position of this paper. Section III introduces our innovative approach, which combines novel graph walks and representation learning methods for RDF-star. Section IV describes the construction of our complex RDF-star graph dataset, featuring multi-leveled nested structures employed in our experimental analysis. Section V describes evaluation tasks, describes the gold standard dataset, and discusses the evaluation results and parameter analysis. Section VI concludes the paper, including a brief summary and future works.

## II. RELATED WORK

### A. KNOWLEDGE GRAPH EMBEDDINGS FOR RDF

Numerous KGE methods have been proposed [18], encompassing graph walk-based, translation-based, and graph neural network methods. This paper focuses on graph walk-based methods applied to RDF graphs. RDF2Vec [8], [9] is a well-known graph walk-based KGE method for RDF graphs. Initially, RDF2Vec generates a sequence set using a random walk. Subsequently, vertices and edges are relabeled employing Weisfeiler-Lehman (WL) graph kernels for RDF [19]. The finalized sequence set serves as input to word2vec [20]. Cochez et al. [21] extended this by incorporating biased random walks for more semantically rich walks. Portisch et al. [22] proposed RDF2Vec Light, a streamlined variant of RDF2Vec that reduces computational complexities by generating vectors only for entities of interest. Moreover, Portisch et al. introduced an order-aware RDF2Vec (RDF2Vec<sub>oa</sub>) [23] using structured word2vec [14], focusing on the original word2vec model’s positional insensitivity. Additionally, they proposed similarity-oriented and relevance-oriented walk methods and identified the advantages of each method [24]. Steenwinckel et al. [25] showed that the WL graph kernel offers little improvements in the context of a single KG with respect to walk embeddings, and proposed five alternative walking strategies.

In this way, numerous KGE methods for RDF have been developed, and most of them were influenced by RDF2Vec. Our approach is positioned as an extension of RDF2Vec. To

**TABLE 1.** Comparison table describing embedding methods for RDF and hyper-relational KGs

	RDF-star2Vec (Ours)	RDF2Vec [8]	RDF2Vec <sub>oa</sub> [23]	m-TransH [26]	NaLP [12]	StarE [11]
<b>Target tasks</b>	Data mining			Link prediction		
<b>Types of KGE</b>	Graph walk			Translation	FCN	GNN
<b>Handling RDF data directly</b>	✓	✓	✓	-	-	-
<b>Handling RDF-star data directly</b>	✓	-	-	-	-	-
<b>Hyper-relational KGE</b>	✓	-	-	✓	✓	✓
<b>Recursive hyper-relational KGE</b>	✓	-	-	-	-	-
<b>QT embedding</b>	✓	-	-	-	-	-
<b>QT-QT relation embedding</b>	✓	-	-	-	-	-

the best of our knowledge, our approach is the first method capable of directly representing recursive hyper-relational KGs described in RDF-star in low-dimensional vector space without any information loss.

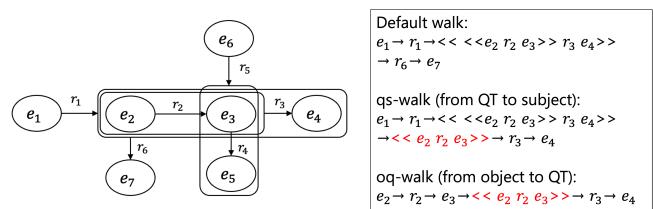
### B. HYPER-RELATIONAL KNOWLEDGE GRAPH EMBEDDINGS

Wen et al. [26] proposed m-TransH, a generalized model of TransH [27] for link prediction of Hyper-relational KGs. However, m-TransH does not consider the relatedness of the components in the same hyper-relation. Guan et al. [12] proposed NaLP, a method to explicitly model the relatedness of the role-value pairs involved in the same hyper-relation. While m-TransH is grounded in a translation-based link prediction model, NaLP employs a fully connected neural network (FCN). Galkin et al. [11] proposed StarE, a graph neural network (GNN)-based link prediction model for hyper-relational KG. StarE extended CompGCN [28] to handle *qualifiers* of Wikidata. However, these approaches focus on link prediction in simple hyper-relational KGs without recursive structures. Therefore, these approaches are different from the purpose of this study. In addition, they are unable to load standard RDF or RDF-star data directly. Note that these studies have generated their benchmark datasets. A comparison table describing the embedding methods for RDF and hyper-relational KGs is presented in Table 1.

Kwan et al. [10] proposed ExtRet, an algorithm designed to convert RDF-star graphs into regular RDF graphs. ExtRet extended the representation of standard RDF Reification to minimize structural information loss to perform link predictions on binary relations. In contrast, our method can generate embeddings directly without converting the structure of the original RDF-star graphs. Therefore, no additional properties or intermediate nodes are generated due to the conversion, and the embeddings can be produced for the original RDF-star without any information loss.

### C. BENCHMARKING DATASETS FOR HYPER-RELATIONAL KNOWLEDGE GRAPH EMBEDDING

A well-known dataset in the realm of hyper-relational KG is Wikidata [29], which contains *qualifiers* to specify re-

**FIGURE 2.** Graph walk methods for RDF-star

lationships and represents supplementary information in a key-value format. WikiPeople [12], a benchmark dataset consisting of information about individuals, is also extracted from Wikidata and serves the purpose of evaluating link prediction models for hyper-relational KGs. JF17K [26], another benchmark, is extracted from Freebase [30]. However, it was pointed out that WikiPeople contains numerous literal values that are conventionally ignored in KGE approaches, and JF17K has been criticized for significant test leakage. As a response, Galkin et al. [11] proposed WD50K as an alternative benchmark dataset designed for the link prediction tasks.

However, existing datasets lack complex structures such as multi-leveled nested QTs and QT-QT relations. Thus, we have developed a complex RDF-star dataset based on KGRC-RDF [15], [16], [31] and integrated its gold standard dataset into GEval [17], thereby enabling evaluations in classification, clustering, entity relatedness, and QT similarity tasks.

## III. RDF-STAR2VEC

### A. GRAPH WALKS

The proposed RDF-star2Vec is a graph walk-based embedding model based on RDF2Vec. Let  $e \in \mathcal{E}$  denote an entity with Uniform Resource Identifier (URI) (i.e., the resource in RDF) and  $r \in \mathcal{R}$  denotes a relation (i.e., the property in RDF). A QT  $q \in \mathcal{Q}$  is represented by a triple of  $q := (n, r, n)$ , where node  $n \in \mathcal{E} \cup \mathcal{Q}$ . Therefore, RDF-star graph is  $\mathcal{G} \subseteq (\mathcal{E} \cup \mathcal{Q}) \times \mathcal{R} \times (\mathcal{E} \cup \mathcal{Q})$ .

An example of an RDF-star graph containing the complex structure used in the following description is displayed in Figure 2. Similar to the default walks in RDF2Vec, the walk path is as  $e_1 \rightarrow r_1 \rightarrow << e_2 r_2 e_3 >> r_3 e_4 >> \rightarrow$

**Algorithm 1** Random walk of RDF-star2Vec

---

**Require:** Root node  $e$ , # of walks  $n$ , Depth  $d$ , Map of QTs  $qts$   
**Ensure:** Walk list  $wl$

- 1:  $wl \leftarrow$  an empty list
- 2: **for**  $currentDepth < d$  **do**
- 3:   **if**  $currentDepth$  is 0 **then**
- 4:      $triples_e^{subj} \leftarrow$  getTriplesInvolvingSubj( $e$ )
- 5:     /\*  $triples_e^{subj}$  means a list of triples which involving  $e$  as subject \*/
- 6:      $qt_e^{obj} \leftarrow$  getRandQtInvolvingObj( $e$ ,  $triples_e^{subj}$ ,  $qts$ )
- 7:     /\*  $qt_e^{obj}$  means a QT which involving  $e$  as object \*/
- 8:      $qt_e^{subj} \leftarrow$  getRandQtInvolvingSubj( $e$ ,  $triples_e^{subj}$ ,  $qts$ )
- 9:      $walk \leftarrow$  an empty list
- 10:    /\* Continue to Algorithm 2 \*/
- 11:     $wl \leftarrow$  QTWalk( $wl$ ,  $walk$ ,  $triples_e^{subj}$ ,  $qt_e^{obj}$ ,  $qt_e^{subj}$ )
- 12:   **else**
- 13:     **for**  $walk \in wl$  **do**
- 14:        $lastObj \leftarrow walk.get(walk.size - 1)$
- 15:       **if**  $walk$  is not QT **then**
- 16:          $lastPred \leftarrow walk.get(walk.size - 2)$
- 17:          $lastSubj \leftarrow walk.get(walk.size - 3)$
- 18:       **end if**
- 19:        $triples_{lastObj}^{subj} \leftarrow$  getTriplesInvolvingSubj( $lastObj$ )
- 20:        $qt_{lastObj}^{obj} \leftarrow$  getQT( $lastSubj$ ,  $lastPred$ ,  $lastObj$ )
- 21:        $qt_{lastObj}^{subj} \leftarrow$  getRandQtInvolvingSubj( $lastObj$ ,  $triples_{lastObj}^{subj}$ ,  $qts$ )
- 22:        $triples_e^{subj} \leftarrow tripes_{lastObj}^{subj}$
- 23:        $qt_e^{obj} \leftarrow qt_{lastObj}^{obj}$
- 24:        $qt_e^{subj} \leftarrow qt_{lastObj}^{subj}$
- 25:       /\* Continue to Algorithm 2 \*/
- 26:        $wl \leftarrow$  QTWalk( $wl$ ,  $walk$ ,  $triples_e^{subj}$ ,  $qt_e^{obj}$ ,  $qt_e^{subj}$ )
- 27:     **end for**
- 28:   **end if**
- 29:   /\* trim the list \*/
- 30:   **while**  $wl.size > n$  **do**
- 31:     remove a random item from  $wl$
- 32:   **end while**
- 33: **end for**

---

$r_6 \rightarrow e_7$ , when starting from  $e_1$ , where  $<<< e_2 r_2 e_3 >>$   $r_3 e_4 >>$  denotes a multi-leveled nested QT. Similar to other entities, this multi-leveled nested QT is treated as a node. Thus, in the default walk, for each node  $n \in \mathcal{E} \cup \mathcal{Q}$  in a given  $\mathcal{G}$ , we generate all sequences  $S_n^d$  of depth  $d$  rooted in the node  $n$ . The  $S_n^d$  represents a set of sequences  $s_n'^d$  expressed in Equation 1.

$$s_n'^d = n, r_{1,j}, n_{1,j}, r_{2,j}, n_{2,j}, \dots, r_{d,j}, n_{d,j} \quad (1)$$

where  $j \in \mathcal{R}(n)$  denotes an element of the preceding node's relationships. Thus, the final sequence set generated by the default walks at depth  $d$  on the single RDF-star graph is  $\bigcup_{n \in \mathcal{E} \cup \mathcal{Q}} S_n^d$ .

As a limitation, the default walks cannot accurately extract the semantics originally expressed by QTs because it cannot walk the compositional entities of QTs. Thus, we propose a new walk method "QT-walk" between the QTs and their compositional entities. Specifically, we propose the following two types of QT-walk.

**Algorithm 2** QT-walk generation

---

**Require:**  $wl$ ,  $walk$ ,  $triples_e^{subj}$ ,  $qt_e^{obj}$ ,  $qt_e^{subj}$ ,  $\alpha$ ,  $\beta$   
**Ensure:**  $wl$

- 1:  $rand_{oq} \leftarrow$  a random number
- 2:  $rand_{qs} \leftarrow$  a random number
- 3:  $newWalk \leftarrow$  a copy of  $walk$
- 4: **if**  $qt_e^{obj}$  is not null AND  $rand_{oq} < \beta$  **then**
- 5:   /\* **oq-walk: from object to QT** \*/
- 6:    $qt \leftarrow qt_e^{obj}.qt$
- 7:   /\*  $qt_e^{obj}.qt$  means  $qt_e^{obj}$  with quotes, i.e., <  $subj$  pred  $obj$  > \*/
- 8:   **if**  $walk$  is empty **then**
- 9:     append  $qt_e^{obj}.obj$  to  $newWalk$
- 10:    /\*  $qt_e^{obj}.obj$  means the object of the  $qt_e^{obj}$  \*/
- 11:   **end if**
- 12:   append  $qt$  to  $newWalk$
- 13:   remove  $walk$  from  $wl$
- 14:   append  $newWalk$  to  $wl$
- 15: **else if**  $qt_e^{subj}$  is not null AND  $rand_{qs} < \alpha$  **then**
- 16:   /\* **qs-walk: from QT to subject** \*/
- 17:   **if**  $walk$  is empty **then**
- 18:     append  $qt_e^{subj}.subj$  to  $newWalk$
- 19:   **end if**
- 20:   append  $qt_e^{subj}.pred$  to  $newWalk$
- 21:   append  $qt_e^{subj}.obj$  to  $newWalk$
- 22:   append  $newWalk$  to  $wl$
- 23: **else**
- 24:   /\* **default walk** \*/
- 25:   remove  $walk$  from  $wl$
- 26:   **for**  $triple \in tripes_e^{subj}$  **do**
- 27:     append predicate of  $triple$  to  $newWalk$
- 28:     append object of  $triple$  to  $newWalk$
- 29:     append  $newWalk$  to  $wl$
- 30:   **end for**
- 31: **end if**
- 32: **end if**

---

- (1) **qs-walk:** walk from a QT to its compositional entity in the subject role
- (2) **oq-walk:** walk from a compositional entity in the object role to the QT

For example, in Figure 2, the qs-walk generates a sequence as  $e_1 \rightarrow r_1 \rightarrow <<< e_2 r_2 e_3 >> r_3 e_4 >> \rightarrow << e_2 r_2 e_3 >> \rightarrow r_3 \rightarrow e_4$  when the starting point is  $e_1$ . The oq-walk generates a sequence as  $e_2 \rightarrow r_2 \rightarrow e_3 \rightarrow << e_2 r_2 e_3 >> \rightarrow r_3 \rightarrow e_4$  when the starting point is  $e_2$ . Thus, in the QT-walk, for a given  $\mathcal{G}$ , for each node  $n \in \mathcal{E} \cup \mathcal{Q}$ , we generate all sequence  $P_n^d$  of depth  $d$  rooted in the node  $n$ . The  $P_n^d$  is a set of sequences  $p_n'^d$  shown in Equation 2.

$$p_n'^d = n, w(n_{1,j}), w(n_{2,j}), \dots, w(n_{d,j}) \quad (2)$$

where  $w(n_{i,j})$  is a QT  $q_{i,j}$ , a triple  $(n_{i,j}^{subj}, r_{i,j}, n_{i,j}^{obj})$ , or a pair  $(r_{i+1,j}, n_{i+1,j})$ . The  $q_{i,j}$  is an expression of a QT with quotes ("<<" and ">>"). The  $(n_{i,j}^{subj}, r_{i,j}, n_{i,j}^{obj})$  represents a compositional triple of a QT. The  $(r_{i+1,j}, n_{i+1,j})$  denotes the pair of predicate and object related to  $n_{i,j}$ . Therefore, the final set of the sequences generated by the QT-walk of depth  $d$  on the single RDF-star graph is  $\bigcup_{n \in \mathcal{E} \cup \mathcal{Q}} P_n^d$ .

Algorithms 1 and 2 describe the graph walk of our method. Specifically, Algorithm 1 represents an algorithm that retrieves candidate QTs and triples for the graph walk. Algorithm 2 is the QT-walk that cases  $w(n_{i,j})$  in Equation 2.

In addition, we introduce two parameters  $\alpha$  and  $\beta$  to set the transition probability. The  $\alpha$  is the transition probability from a QT to its compositional entity in the subject role. The  $\beta$  is the transition probability from a compositional entity in the object role to the QT. The former walks deeply into the nested structure, while the latter walks out of the nested structure. The parameters  $\alpha$  and  $\beta$  are set manually in the range  $\alpha \in (0, 1]$  and  $\beta \in (0, 1]$ , respectively. If both the qs-walk and oq-walk transitions are possible, priority is given to the oq-walk.

We also introduce mid walk [22] to RDF-star2Vec. Instead of starting random walks at all entities of interest, it is randomly decided for each depth iteration whether to go backward, i.e., to one of the node's predecessors, or forwards, i.e., to the node's successors. The mid walk of the proposed method is explained in Algorithm 3.

### B. REPRESENTATION LEARNING

DeepWalk [32], node2vec [33], RDF2Vec [8] and other representative graph walk-based KGE methods use word2vec [20], a neural network-based representation learning method of words, to encode entities and relations into distributed representations from the generated sequence set. Word2vec offers two principal learning algorithms: the continuous bag-of-words (CBOW) and skip-gram models. Our proposed method utilizes the skip-gram model for learning distributed representations from sequence sets, as empirical evidence suggests that the skip-gram model outperforms CBOW model in the context of RDF2Vec. The skip-gram model aims to maximize the average log probability denoted by Equation 3, given a sequence of training words  $w_1, w_2, \dots, w_t$ .

$$\frac{1}{T} \sum_{t=1}^T \sum_j \log(p(w_{t+j}|w_t)) \quad (3)$$

where  $T$  indicates the number of words and  $w_{t+j}$  ( $-c \leq j \leq c$ ) represents the words appearing in context window  $c$ . The probability  $p(w_o|w_i)$  of an input word  $w_i$  and an output word  $w_o$  is calculated using the Softmax function in Equation 4.

$$p(w_o|w_i) = \frac{\exp(v_{w_o}^T \cdot v_{w_i})}{\sum_{w \in W} \exp(v_w^T \cdot v_{w_i})} \quad (4)$$

where  $v_w$  and  $v'_w$  represent the input and output vectors of the word  $w$ , and  $W$  is the word set.

Portisch et al. demonstrated a partial enhancement in RDF2Vec's performance by incorporating structured word2vec [14], which takes word order into account [23]. Therefore, in the proposed method, we employ structured word2vec, specifically the structured skip-gram model, expecting to improve the embedding performance of the context of the QT-walk. Figure 3(b) illustrates the architecture of

---

### Algorithm 3 Mid walk of RDF-star2Vec

---

**Require:** Node  $e$ , # of walks  $n$ , Depth  $d$ , Map of QTs  $qts$   
**Ensure:** Walks list  $wl$

```

1:  $wl \leftarrow$  an empty list
2:  $np \leftarrow e$  /* next predecessor */
3:  $ns \leftarrow e$  /* next successor */
4: while  $wl.size < n$  do
5:   while  $currentDepth < d$  do
6:      $rand_{oq} \leftarrow$  a random number
7:      $rand_{qs} \leftarrow$  a random number
8:      $zo \leftarrow$  randomPick( $\{0, 1\}$ )
9:     if  $zo$  is 0 then
10:      /* next predecessor */
11:       $tpls_{np}^{obj} \leftarrow$  getTriplesInvolvingObj( $np$ )
12:       $qt_{np}^{obj} \leftarrow$  getRandQtInvolvingObj( $np$ ,  $tpls_{np}^{obj}$ ,  $qts$ )
13:      if  $qt_{np}^{obj}$  is not null AND  $rand_{oq} < \beta$  then
14:        /* oq-walk */
15:        append  $qt_{np}^{obj}.obj$  to the beginning of  $walk$ 
16:        append  $qt_{np}^{obj}.pred$  to the beginning of  $walk$ 
17:        append  $qt_{np}^{obj}.subj$  to the beginning of  $walk$ 
18:         $currentDepth++$ 
19:         $np \leftarrow qt_{np}^{obj}.subj$ 
20:      else
21:         $triple \leftarrow$  getRandomTriple( $tpls_{np}^{obj}$ )
22:        append  $triple$  to the beginning of  $walk$ 
23:         $np \leftarrow$  the subject of  $triple$ 
24:      end if
25:    else
26:      /* next successor */
27:      if  $walk.size \geq 3$  then
28:         $lastPred \leftarrow walk.get(walk.size - 2)$ 
29:         $lastSubj \leftarrow walk.get(walk.size - 3)$ 
30:      end if
31:       $tpls_{ns}^{subj} \leftarrow$  getTplsInvolvingSubj( $ns$ )
32:       $qt_{ns}^{subj} \leftarrow$  getRandQtInvolvingSubj( $ns$ ,  $tpls_{ns}^{subj}$ ,  $qts$ )
33:      if  $qt_{ns}^{subj}$  is not null AND  $rand_{qs} < \alpha$  then
34:        /* qs-walk */
35:        append  $qt_{ns}^{subj}.subj$  to the end of  $walk$ 
36:        append  $qt_{ns}^{subj}.pred$  to the end of  $walk$ 
37:        append  $qt_{ns}^{subj}.obj$  to the end of  $walk$ 
38:         $ns \leftarrow qt_{ns}^{subj}.obj$ 
39:      else
40:         $triple \leftarrow$  getRandomTriple( $tpls_{ns}^{subj}$ )
41:        append  $triple$  to the beginning of  $walk$ 
42:         $ns \leftarrow$  the subject of  $triple$ 
43:      end if
44:    end if
45:  end while
46:  append  $walk$  to  $wl$ 
47: end while

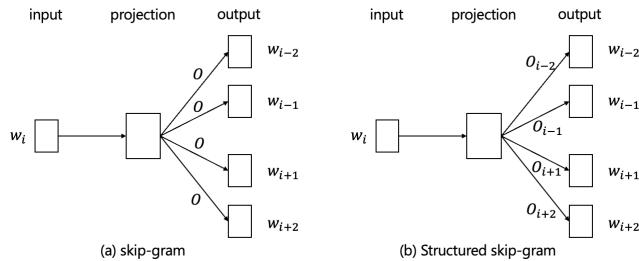
```

---

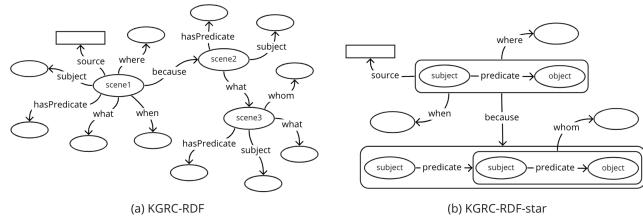
the structured skip-gram. The classic skip-gram uses a single output matrix  $O \in \Re^{|W| \times d}$  ( $d$ =dimension) to predict every contextual word  $w_{i-c}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+c}$ , given the embeddings of the center word  $w_i$ ; in contrast, the structured skip-gram uses different output matrix  $O_{o-i}$  for a specific relative position to the center word.

### IV. DATASET

Given that existing hyper-relational KG datasets such as WikiPeople, JF17K, and WD50K lack complex structures,



**FIGURE 3.** Architecture of the skip-gram [20] and structured skip-gram model [14]



**FIGURE 4.** Data structures of KGRC-RDF and KGRC-RDF-star

they are unsuitable for evaluating the proposed method. Thus, we provide a dataset KGRC-RDF-star that is available for benchmarking the embeddings of complex RDF-star graphs by converting the KGRC-RDF<sup>6</sup> [15], [31] to RDF-star data, which is provided for International Knowledge Graph Reasoning Challenge (IKGRC)<sup>7</sup>.

The original KGRC-RDF is a set of eight KGs built based on Sherlock Holmes mystery stories and has been published as an RDF dataset for benchmarking XAI systems that can provide reasons for its decisions. Figure 4(a) shows the data structure of KGRC-RDF. The KGRC-RDF contains detailed descriptions of “who, what, when, where, why, and how (5W1H)” information along with the storyline and has a rich variation of entities as values of 5W1H. Several approaches [31], [34], [35] were proposed to estimate the criminal from the KGRC-RDF and present a convincing explanation by KGE and logical reasoning technologies.

We converted the KGRC-RDF graphs to the RDF-star graphs, as shown in Figure 4(b), and provided it as a dataset for evaluating the RDF-star embeddings. For example, the scene “Julia met a lieutenant commander two years ago at Harrow.” was described in the KGRC-RDF as follows.

```

1 @prefix kgc: <http://kgc.knowledge-graph.jp/
    ontology/kgc.owl#> .
2 @prefix kdsb: <http://kgc.knowledge-graph.jp/
    data/SpecleckBand/> .
3 kdsb:36 a kgc:Situation ;
4   kgc:hasPredicate kdp:meet ;
5   kgc:subject kdsb:Julia ;
6   kgc:then kdsb:37 ;
7   kgc:when kdsb:2_years_ago ;

```

The above description was converted as follows.

<sup>6</sup><https://github.com/KnowledgeGraphJapan/KGRC-RDF/tree/ikgrc2023>  
<sup>7</sup><https://ikgrc.org/>

**TABLE 2.** Statistics of the KGRC-RDF-star

	Class	17
	Instance	9,524
	Property	640
Triple	Standard triple	14,180
	Single-nested QT	9,765
	Double-nested QT	6,409
	Triple-nested QT	695
	Quadruple-nested QT	43
	Total	31,092

```

1 << kdsb:Julia kdp:meet kdsb:
    lieutenant_commander >>
2 a kgc:Situation ;
3 kgc:then << kdsb:Julia kdp:engage kdsb:
    lieutenant_commander >> ;
4 kgc:when kdsb:2_years_ago ;
5 kgc:where kdsb:Harrow .

```

It is necessary to specify an object of a triple explicitly when converting from the KGRC-RDF to the KGRC-RDF-star. We set the priority for the object selection as follows: *what* > *whom* > *where* > *on* > *to* > *from*. If either subject or object did not exist, owl:Nothing was substituted.

QTs are unique for each combination of subject *s*, predicate *p*, and object *o*, and there is no URI and ID for identifying a QT. However, the same *s*, *p*, and *o* combinations might occur in different scenes when the KGRC-RDF is converted to the KGRC-RDF-star. It is necessary to distinguish these QTs and assign different metadata to them. Therefore, we solved this issue by assigning a unique ID to each QT and nested these triples as a QT as follows: << << *s p o* >> *id val* >> *pt ot*.

Table 2 provides a statistical overview of the KGRC-RDF-star. Notably, these statistics exclude the nesting introduced to address the issue of QT uniqueness. The finalized dataset has been made publicly accessible via GitHub<sup>8</sup>.

## V. EVALUATION

### A. EVALUATION TASKS

For the evaluation of RDF-star embeddings generated through our proposed method, we focus on classification, clustering, entity relatedness, and QT similarity tasks. We selected RDF2Vec and RDF2vec<sub>oa</sub> as the baseline methods of KGE. Here, we extended jRDF2Vec<sup>9</sup> to enable to generate sequences of RDF-star data, since RDF2vec cannot load RDF-star data. We employ an evaluation framework for graph embedding, GEval [17], to evaluate the performance of each task. Although the GEval provides the entity list of DBpedia [36] as the default gold standard dataset, DBpedia is not RDF-star data and is not appropriate for the evaluation in this experiment. Thus, we constructed gold standard datasets

<sup>8</sup><https://github.com/aistairc/KGRC-RDF-star>

<sup>9</sup><https://github.com/dwslab/jRDF2Vec>

**TABLE 3.** Details of the classification task

	<b>Dataset</b>	<b>Semantics of classes</b>	<b>Classes</b>	<b>Size</b>
<b>INPUT</b>	PersonObjectPlace	Type of entities	3	540
	QT900	Type of quoted triples	3	900
<b>Model</b>	<b>Model</b>	<b>Configuration</b>		-
	Naive Bayes			-
	C4.5 decision tree			-
	k-NN			k=3
<b>OUTPUT</b>	SVM	$C \in \{10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3\}$		
	<b>Metric</b>	<b>Range</b>	<b>Optimum</b>	
	Accuracy	[0,1]	Highest	

based on KGRC-RDF-star to support benchmarking of the complex RDF-star embeddings and incorporated them into the GEval, and then published them on GitHub<sup>10</sup>. In the following sections, we describe the details of each task and how to create gold standard datasets.

### 1) Classification

The classification task learns training data for entity-label pairs and estimates the labels given unknown entities. Table V-A1 shows the details of the classification task and the gold standard datasets. These gold standard datasets have been designed to quantitatively evaluate whether each entity and QT has represented suitable feature vectors that contribute to the classification task. The dataset PersonObjectPlace comprises pairs of entities and labels, with labels classified into *Person*, *Object*, and *Place*. It is an unbiased dataset randomly extracted from the KGRC-RDF-star using SPARQL queries and designed to have an equal number of each class. Similarly, QT900 contains pairs of QTs and labels, which are classified into *Situation*, *Statement*, and *Thought*, and it is also an unbiased dataset designed to have an equal number of each class. In the classification task, the above values of the types are removed from the source RDF-star dataset when generating embeddings to exclude the potential ground truth information from the feature vectors. Finally, the results are calculated using 10-fold cross-validation.

### 2) Clustering

In the clustering task, clusters are generated from the embeddings using unsupervised methods, and performance is evaluated by comparing the clusters to the gold standard dataset. Table 4 shows the details of the clustering task. The gold standard datasets are the same ones used in the classification task.

### 3) Entity relatedness

In the entity relatedness task, we assume that two entities are related if they often appear in the same context, as in prior

**TABLE 4.** Details of the clustering task

	<b>Dataset</b>	
<b>INPUT</b>	same as the Classification task	
<b>Model</b>	<b>Model</b>	<b>Configuration</b>
	Agglomerative Clustering	similarity_metric
	Ward Hierarchical Clustering	similarity_metric
	DBscan	similarity_metric
<b>OUTPUT</b>	k-Means	-
	<b>Metric</b>	<b>Range</b>
	adjusted rand score	[-1,1]
	adjusted mutual info score	[0,1]
	Fowlkes Mallow index	[0,1]
	v_measure score	[0,1]
	homogeneity score	[0,1]
	completeness score	[0,1]
	accuracy	[0,1]
		Highest

studies [9], [17]. Table 5 shows the details of the entity relatedness task. The kgrc\_entity\_relatedness is a set of 21 person entities extracted from the KGRC-RDF-star and 10 entities related to each person, and the related entities are sorted by their relatedness. This gold standard dataset was created as follows, referring to the methodology of KORE [37].

- (1) Twenty-one person entities are extracted as seed entities from the KGRC-RDF-star.
- (2) Ten candidates are extracted from sorted lists of entities that co-occur with the seed entity in the same scenes (210 in total).
- (3) All possible comparisons of the 10 candidates with respect to their seed are created (945 in total).
- (4) Amazon Mechanical Turk (MTurk)<sup>11</sup> workers are asked which of the given two entities is more related to the seed entity. Ten workers answer each question.
- (5) Ten comparison pairs are created as the gold standard by authors to improve the quality of the crowdsourcing

<sup>10</sup><https://github.com/aistairc/GEval-forKGRC-RDF-star/>

<sup>11</sup><https://www.mturk.com/>

**TABLE 5.** Details of the entity relatedness task

INPUT	Dataset	Structure	Size
	kgrc_entity_relatedness	Person entity with a sorted list of 10 related entities	210
Model	Model	Config	
	Create sorted lists of similarity scores between each person entity and related entities	similarity_metric	
OUTPUT	Metric	Range	Interpretation
	Kendall's Tau (Kendall Rank Correlation Coefficient)	[-1,1]	Values close to 1: correlation Values close to 0: no correlation

task. Spam workers are removed based on the answers for these pairs.

- (6) All the comparisons are aggregated into a single confidence that one entity is more (or equally) related to the seed.
- (7) The ten candidate entities are subsequently ranked according to these confidence values.

The evaluation model is a simple algorithm that sorts the list of candidate entities based on the similarity scores between the vectors of the seed entity and the candidate entities. Finally, the output list is compared to the gold standard dataset using Kendall's rank correlation coefficient.

#### 4) QT similarity

A QT in the KGRC-RDF-star corresponds to a scene in KGRC-RDF. We assume that the vectors of similar QTs are located nearby in the embedding space by adequately learning the semantic relations between QTs' neighbors and between the compositional entities of the QTs. Thus, this task aims to verify whether the embedding vectors of the QTs contain semantic and contextual similarity equivalent to human judgment. Table 6 shows the details of the QT similarity task. QT50 was created through the following procedures, using the methodology of LP50 [38].

- (1) Fifty QTs and the scene descriptions corresponding to the QTs are extracted from KGRC-RDF-star and KGRC.
- (2) All possible pairs of the 50 QTs are created (1,225 in total).
- (3) MTurk workers judge the similarity on a five-point scale (with one indicating "not similar at all" and five indicating "quite similar"). 10 workers answer each question.
- (4) Five pairs are created as the gold standard by authors to improve the quality of the crowdsourcing task. Spam workers are removed based on the answers for these pairs.
- (5) The average similarity scores of each pair are calculated.

## B. EVALUATION RESULTS

Table 7 shows the evaluation results. We used cosine similarity as a similarity metric. We preliminarily compared random

walk and mid walk in RDF-star2Vec and found that the mid walk performed better than the random walk. Therefore, we adopted the mid walk in all of our evaluation experiments. We set the following hyperparameters: depth is 8, number of walks per entity is 100, window size is 5, dimension is 100,  $\alpha$  is 0.5, and  $\beta$  is 0.5. The proposed method outperformed RDF2Vec and RDF2Vec<sub>oa</sub> on more than half of the tasks. Specifically, RDF-star2Vec with the structured skip-gram (*oa*) achieved the highest accuracy in the classification task of the PersonObjectPlace dataset. Conversely, all methods achieved good performance in the classification task of the QT900 dataset, and RDF2Vec<sub>oa</sub> slightly outperformed RDF-star2Vec. This difference is slight enough to require a statistical test. To evaluate the significance between the baseline and RDF-star2Vec<sub>oa</sub>, we performed a Wilcoxon signed rank test based on the results of 10-fold cross-validation using the classification model with the best accuracy for each method (i.e., Support Vector Machine (SVM)). We used a standard significance level of  $p < 0.05$ . The result showed that the RDF-star2Vec<sub>oa</sub> is significantly more accurate (p-value = 0.00758) than the baseline in the classification task of the PersonObjectPlace dataset. Conversely, the baseline outperformed the RDF-star2Vec<sub>oa</sub> by 0.002 for the classification of the QT900 dataset. However, a Wilcoxon signed rank test at a significance level of  $p < 0.05$  revealed no significant difference in accuracy in regarding the classification task of the QT900 between the baseline and the RDF-star2Vec<sub>oa</sub> (p-value = 0.131).

RDF-star2Vec<sub>oa</sub> achieved the highest accuracy in the PersonObjectPlace dataset in the clustering task. Conversely, the classic RDF2Vec achieved the highest accuracy in the QT900 dataset. Our proposed method is inclined to walk from a QT to its compositional entities rather than to the rdf:type of the QT. In contrast, RDF2Vec frequently walks to the rdf:type, which corresponds to the cluster information in the gold standard dataset. As a result, the clusters generated by the proposed method deviated from those of the gold standard dataset.

In the entity relatedness task, we observed the highest correlation when using RDF-star2Vec with the classic skip-gram ( $p = 0.0166$ ;  $p < 0.05$  was taken to indicate statistical sig-

**TABLE 6.** Details of the QT similarity task

INPUT	Dataset	Structure	Size
	QT50	QT1 QT2 avg	50 QTs
Model	Model	Configuration	
	similarity(QT1, QT2)	similarity_metric	
OUTPUT	Metric	Range	Interpretation
	Pearson correlation (P <sub>cor</sub> )	[-1,1]	Close to 1: correlation Close to 0: no correlation
	Spearman correlation (S <sub>cor</sub> )	[-1,1]	Close to 1: correlation Close to 0: no correlation
	Harmonic mean of P <sub>cor</sub> and S <sub>cor</sub>	[-1,1]	Close to 1: correlation Close to 0: no correlation

**TABLE 7.** Evaluation results (parameter settings: depth=8, walks\_per\_entity=100, window=5, dimension=100,  $\alpha=0.5$ ,  $\beta=0.5$ )

Task	Metric	Dataset	RDF2vec [8]		RDF-star2Vec	
			classic	oa [23]	classic	oa
Classification	Accuracy	PersonObjectPlace	0.744	0.800	0.759	<b>0.822</b>
	Accuracy	QT900	0.968	<b>0.976</b>	0.971	0.974
Clustering	Accuracy	PersonObjectPlace	0.557	0.751	0.477	<b>0.870</b>
	Accuracy	QT900	<b>0.708</b>	0.690	0.435	0.492
Entity Relatedness	Kendall's Tau	kgrc_entity_relatedness	0.422	0.466	<b>0.600</b>	0.555
QT Similarity	P <sub>cor</sub>	QT50	0.0325	0.0299	<b>0.116</b>	0.100

nificance). The embedding of the proposed method reflects entity relatedness more than the existing methods because the proposed method can place a QT and its compositional entities close to each other in the generated sequences. For instance, when two individuals appear in the same context, they serve as the QT's compositional subject or object entities and share metadata such as time, place, reason, and related scenes. Therefore, the proposed method was able to place these entities (i.e., entities of N-ary relations represented by RDF-star) close to each other in the generated sequence. Consequently, the skip-gram was able to embed entities considering the entity relatedness.

In the QT similarity task, none of the methods performed well. However, the results of the proposed method very slightly correlated with the gold standard dataset.

These results indicate that our proposed method is especially suitable for the classification and clustering tasks of normal entities in RDF-star datasets.

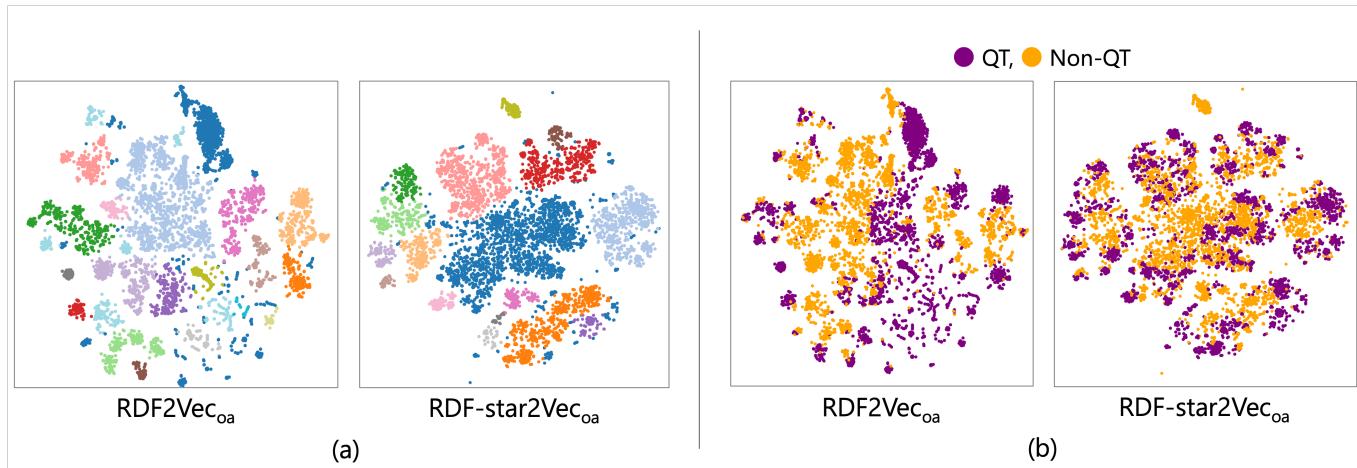
To summarize, RDF-star2Vec<sub>oa</sub> significantly outperformed the baseline in four of the six tasks, was inferior in one task, and equaled the baseline in another. Therefore, when considering all tasks, RDF-star2Vec<sub>oa</sub> overall outperforms the baseline.

### C. VISUALIZATION

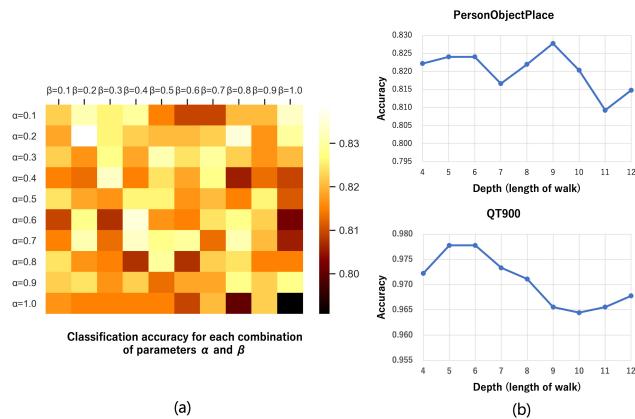
We applied the t-SNE [39] to the 100-dimensional embedding vectors to visualize the results as a two-dimensional plot. Figure 5 shows a comparison between the embedding vectors of baseline (RDF2Vec<sub>oa</sub>) and RDF-star2Vec<sub>oa</sub>. We used the same hyperparameters as in the evaluation experiments in Section V-B.

Figure 5 (a) shows the clustering results using the DBSCAN [40] after reducing the 100-dimensional embedding vectors to two dimensions using t-SNE [39]. DBSCAN, a density-based clustering algorithm, automatically determines the number of clusters. In this visualization, we configured the parameters as follows: the neighborhood distance is 60, and the minimum number of samples in a neighborhood for a core point is 6. In the RDF2Vec<sub>oa</sub>, 20 clusters were formed with finely distributed nodes. In addition, some clusters of the same color were scattered in different locations (e.g., light blue). Conversely, in the RDF-star2Vec<sub>oa</sub>, 17 clusters were formed with fewer instances of noise. Therefore, the RDF-star2Vec's embeddings are better for visualizing cluster characteristics than the RDF2Vec.

Figure 5 (b) shows the embedding vectors, color-coded to distinguish QTs from other nodes. In the RDF2Vec<sub>oa</sub>, there is a polarization between QT and others. Since the RDF2Vec cannot walk the entities constituting a QT, the internal se-



**FIGURE 5.** Visualization results of embeddings of baseline ( $\text{RDF2Vec}_{\text{oa}}$ ) and  $\text{RDF-star2Vec}$ : (a) is the result of clustering using DBSCAN for 100-dimensional embeddings of all nodes, compressed to two dimensions using t-SNE, and (b) is color-coded to distinguish QTs from others.



**FIGURE 6.** Parameter analysis for classification based on the  $\text{RDF-star2Vec}$  embeddings

mantics of the QT are not reflected in the embedding results. This polarization hinders any meaningful analysis of the relationships between QTs and normal entities. In contrast,  $\text{RDF-star2Vec}_{\text{oa}}$  demonstrates no such polarization; it allows for the formation of mixed clusters containing both QTs and normal entities. The  $\text{RDF-star2Vec}$  can visualize QTs and normal entities closely since the internal semantics of the QTs are reflected in the embedding results. Therefore, from this visualization results, we can analyze the relationships between QTs and normal entities as well as the meaningful clusters they form.

#### D. PARAMETER ANALYSIS AND DISCUSSION

We conducted more experiments by changing parameters to analyze and discuss the characteristics of  $\text{RDF-star2Vec}$ . Figure 6(a) shows the accuracy of the classification task for the PersonObjectPlace dataset for each combination of parameters  $\alpha$  (i.e., probability of qs-walk) and  $\beta$  (i.e., probability of oq-walk). Here, the structured skip-gram was used

for representation learning. The results show that  $\alpha = 0.2$  and  $\beta = 0.2$  are the optimal combination, and  $\alpha = 1.0$  and  $\beta = 1.0$  are the worst combination. Since qs-walk and oq-walk are always performed when  $\alpha = 1.0$  and  $\beta = 1.0$ , walks that loop on the same QT frequently occurs as follows:  $q_1 \rightarrow e_1 \rightarrow r_1 \rightarrow e_2 \rightarrow q_1 \rightarrow e_1 \rightarrow r_1 \rightarrow \dots \rightarrow q_1$ . Therefore, it is difficult to adequately learn the representations of entities by using skip-gram because the variety of the generated sequences is reduced.

If  $\alpha = 0$  and  $\beta = 0$ , it is equal to  $\text{RDF2Vec}^{12}$ , which generates the sequence described in Equation 1. In KGRC-RDF-star, the variety of neighbors of normal entities is limited since the  $\text{RDF2Vec}$ 's default walking strategy transition from QT to only metadata, such as "where," "when," "whom," "then," and "because." In contrast, our proposed method probabilistically transitions between QTs and their compositional entities. Hence, the proposed method can probabilistically place QTs before or after the normal entities in the generated sequences. Consequently, we observed that our proposed method improved the embedding performance as we assumed. We also found that values of  $\alpha$  and  $\beta$  less than or equal to 0.5 are suitable since excessive QT-walk leads to poor performance.

Figure 6(b) shows the accuracy of the classification tasks when changing the depth of the walks ( $\alpha = 0.5$ ,  $\beta = 0.5$ ). The figure shows that the large depth of the walks decreases accuracy. The deeper the walks in  $\text{RDF-star2Vec}$ , the greater the number of transitions between QTs and their compositional entities, and the loops will occur as described above. Therefore, such excessive QT-walk affected the accuracy.

Several issues remain to be addressed to improve embedding performance. For example, qs-walk and oq-walk implemented in this study can be further classified in detail. Specifically, the qs-walk transitions from a QT to its compositional subject entity and then walk to the object in the

<sup>12</sup>Note that this is an extended version of the original  $\text{RDF2Vec}$  to walk on RDF-star graphs but does not use QT-walk.

same QT since we emphasize the context of the QT-walk. However, in KGRC-RDF-star, it is also possible to walk from this subject entity to other entities that are not components of the QT. We can also consider another version of oq-walk that ignores the context. Although we did not implement such walks that ignore the context of the QT-walk, it is possible to break away from the loop of the QT-walk using them. In addition, it is also possible to implement qo-walk, which transitions from a QT to its compositional object entity, and sq-walk, which transitions from a subject entity to a QT. In future work, we will implement new parameters to allow these walks. Furthermore, we will incorporate other walk flavors [24], [25].

### E. LIMITATIONS AND POTENTIAL BIASES OF DATASETS

We constructed KGRC-RDF-star based on KGRC-RDF. The KGRC-RDF consists of eight KGs constructed based on eight mystery stories. Although the mystery stories are closed worlds, they are datasets rich in diversity, with a wide variety of entities such as characters, animals, places, objects, events, and statements, and containing various relationships such as actions, types, person relationships, and causal relationships. Thus, we consider the KGRC-RDF and KGRC-RDF-star to be generalized datasets compared to domain-specific datasets such as biomedical KGs.

According to Kozaki et al. [16], there is a bias in the number of some properties among the eight KGs in KGRC-RDF. Since KGRC-RDF-star is based on these KGs, there is also a bias in the number of some properties among the novels. However, this bias does not directly affect the labels in our gold standard dataset.

For instance, our PersonObjectPlace dataset consists of 540 entities with cluster labels based on its `rdf:type` information (# of Person=180, # of Object=180, # of Place=180). Similarly, the QT900 dataset consists of 900 QTs with cluster labels based on its `rdf:type` information (# of Scene=300, # of Situation=300, # of Thought=300). Therefore, there is no bias in the number of correct labels. However, the variation in the novels from which the gold standard data was extracted is slightly biased. This is because there is a bias in the number of characters and places in different novels. In other words, each eight KG has its own characteristics, and we consider that evaluation experiments can be conducted according to the purpose by restricting the number of KGs used in an experiment.

The experimental results are somewhat generalized since this study used the eight KGs. We expect the future development of an even more robust benchmark dataset by including additional data.

### VI. CONCLUSION

In this paper, we proposed RDF-star2Vec, a novel graph walk-based KGE method, which directly represents RDF-star graphs with complex structures, such as multi-leveled nested QTs and QT-QT relations, in low-dimensional vector

space without any information loss. The approach overcomes the challenge of embedding RDF-star's QTs in vector space by placing related entities closer together in generated sequences.

In addition, we constructed an RDF-star dataset containing complex structures based on KGRC-RDF, and incorporated four gold standard datasets based on it into GEval to enable benchmarking RDF-star embeddings. This framework facilitates evaluating the performance of future KGE methods for RDF-star.

Our proposed method demonstrated significantly better performance than existing methods in tasks such as classification, clustering, entity relatedness, and QT similarity tasks. Future work includes implementing other possible walks, optimizing hyperparameters, and constructing an RDF-star dataset to evaluate additional tasks such as regression, semantic analogy, and link prediction.

It is expected that many N-ary relation graphs (i.e., Hyper-relational KGs) will be published in RDF-star format in the future. We believe that the proposed method will be used as data mining tasks for RDF-star in various domains in the future.

### ACKNOWLEDGMENT

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and JSPS KAKENHI Grant Numbers JP19H04168, JP22K18008, and JP23H03688.

### REFERENCES

- [1] O. Hartig, “RDF\* and SPARQL\*: An Alternative Approach to Annotate Statements in RDF,” in *ISWC (Posters, Demos & Industry Tracks)*, 2017, pp. 1–4.
- [2] B. Kasenbach, A. Lehnert, and G. Loh, “Use case: Ontologies and rdf-star for knowledge management,” in *The Semantic Web: ESWC 2021 Satellite Events*, R. Verborgh, A. Dimou, A. Hogan, C. d’Amato, I. Tiddi, A. Bröring, S. Mayer, F. Ongena, R. Tommasini, and M. Alam, Eds. Cham: Springer International Publishing, 2021, pp. 254–260.
- [3] A. Rivas and M.-E. Vidal, “Capturing knowledge about drug-drug interactions to enhance treatment effectiveness,” in *Proceedings of the 11th on Knowledge Capture Conference*, ser. K-CAP ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 33–40.
- [4] C. H.-J. Braun and T. Käfer, “Self-verifying web resource representations using solid, rdf-star and signed uris,” in *The Semantic Web: ESWC 2022 Satellite Events*, P. Groth, A. Rula, J. Schneider, I. Tiddi, E. Simperl, P. Alexopoulos, R. Hoekstra, M. Alam, A. Dimou, and M. Tamper, Eds. Cham: Springer International Publishing, 2022, pp. 138–142.
- [5] T. Delva, J. Arenas-Guerrero, A. Iglesias-Molina, O. Corcho, D. Chaves-Fraga, and A. Dimou, “RML-star: A declarative mapping language for RDF-star generation,” in *ISWC2021, the International Semantic Web Conference*, 2021, pp. 1–5.
- [6] L. Sundqvist, “Extending vkg systems with rdf-star support,” Master’s thesis, Free University of Bozen-Bolzano. [Online]. Available: <https://ontop-vkg.org/publications/2022-sundqvist-rdf-star-ontop-msc-thesis.pdf>
- [7] G. Abuoda, D. Dell’Aglio, A. Keen, and K. Hose, “Transforming RDF-star to property graphs: A preliminary analysis of transformation approaches,” in *QuWeDa 2022: 6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs*, ser. CEUR Workshop Proceedings, vol. 3279. CEUR Workshop Proceedings, Nov. 2022, pp. 17–32.
- [8] P. Ristoski and H. Paulheim, “RDF2Vec: RDF Graph Embeddings for Data Mining,” in *The Semantic Web – ISWC 2016*, ser. Lecture Notes in Computer Science, P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch,

- F. Lecue, F. Flöck, and Y. Gil, Eds. Cham: Springer International Publishing, 2016, pp. 498–514.
- [9] P. Ristoski, J. Rosati, T. Di Noia, R. De Leone, and H. Paulheim, “RDF2Vec: RDF graph embeddings and their applications,” *Semantic Web*, vol. 10, no. 4, pp. 721–752, 2019.
- [10] L. Kwan, P. G. Omran, and A. Haller, “RDF\* Knowledge Graph Completion by Translation,” in *CEUR Workshop Proceedings*, ser. ISWC 2022 Posters, Demos, and Industry Tracks, vol. 2721, 2022, pp. 1–5.
- [11] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann, “Message passing for hyper-relational knowledge graphs,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 7346–7359.
- [12] S. Guan, X. Jin, Y. Wang, and X. Cheng, “Link prediction on n-ary relational data,” in *Proceedings of the 28th International Conference on World Wide Web (WWW’19)*, 2019, pp. 583–593.
- [13] P. Rosso, D. Yang, and P. Cudré-Mauroux, “Beyond triplets: Hyper-relational knowledge graph embedding for link prediction,” in *Proceedings of The Web Conference 2020*, ser. WWW ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1885–1896.
- [14] W. Ling, C. Dyer, A. W. Black, and I. Trancoso, “Two/Too Simple Adaptations of Word2Vec for Syntax Problems,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 1299–1304.
- [15] T. Kawamura, S. Egami, K. Matsushita, T. Ugai, K. Fukuda, and K. Kozaki, “Contextualized Scene Knowledge Graphs for XAI Benchmarking,” in *Proceedings of the 11th International Joint Conference on Knowledge Graphs*, ser. IJCKG’22, to appear, pp. 1–9. [Online]. Available: [http://ijckg.org/2022/papers/IJCKG\\_2022\\_paper\\_6603.pdf](http://ijckg.org/2022/papers/IJCKG_2022_paper_6603.pdf)
- [16] K. Kozaki, S. Egami, K. Matsushita, T. Ugai, T. Kawamura, and K. Fukuda, “Datasets of Mystery Stories for Knowledge Graph Reasoning Challenge,” *Joint Proceedings of the ESWC 2023 Workshops and Tutorials co-located with 20th European Semantic Web Conference (ESWC 2023)*, vol. 3443, pp. 1–15, 2023.
- [17] M. A. Pellegrino, A. Altabba, M. Garofalo, P. Ristoski, and M. Cochez, “GEval: A Modular and Extensible Evaluation Framework for Graph Embedding Techniques,” in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, and M. Cochez, Eds. Cham: Springer International Publishing, 2020, pp. 565–582.
- [18] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.
- [19] G. K. D. de Vries, “A Fast Approximation of the Weisfeiler-Lehman Graph Kernel for RDF Data,” in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, Eds. Berlin, Heidelberg: Springer, 2013, pp. 606–621.
- [20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [21] M. Cochez, P. Ristoski, S. P. Ponzetto, and H. Paulheim, “Biased graph walks for rdf graph embeddings,” ser. WIMS ’17. Association for Computing Machinery, 2017.
- [22] J. Portisch, M. Hladík, and H. Paulheim, “RDF2Vec Light – a lightweight approach for knowledge graph embeddings,” in *CEUR Workshop Proceedings*, ser. ISWC 2020 Posters, Demos, and Industry Tracks, vol. 2721, 2020, pp. 79–84.
- [23] J. Portisch and H. Paulheim, “Putting RDF2vec in Order,” in *CEUR Workshop Proceedings*, ser. ISWC 2021 Posters, Demos, and Industry Tracks, vol. 2980, 2021, pp. 1–5.
- [24] J. Portisch and H. Paulheim, “Walk This Way! Entity Walks and Property Walks for RDF2vec,” in *The Semantic Web: ESWC 2022 Satellite Events*, P. Groth, A. Rula, J. Schneider, I. Tiddi, E. Simperl, P. Alexopoulos, R. Hoekstra, M. Alam, A. Dimou, and M. Tamper, Eds. Cham: Springer International Publishing, 2022, pp. 133–137.
- [25] B. Steenwinckel, G. Vandewiele, P. Bonte, M. Weyns, H. Paulheim, P. Ristoski, F. De Turck, and F. Ongenae, “Walk extraction strategies for node embeddings with RDF2Vec in knowledge graphs,” in *Database and Expert Systems Applications - DEXA 2021 Workshops*, G. Kotsis, A. M. Tjoa, I. Khalil, B. Moser, A. Mashkoor, J. Sametinger, A. Fensel, J. Martínez-Gil, L. Fischer, G. Czech, F. Sobieczky, and S. Khan, Eds. Cham: Springer International Publishing, 2021, pp. 70–80.
- [26] J. Wen, J. Li, Y. Mao, S. Chen, and R. Zhang, “On the representation and embedding of knowledge bases beyond binary relations,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI’16. AAAI Press, 2016, p. 1300–1307.
- [27] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge Graph Embedding by Translating on Hyperplanes,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014, number: 1. [Accessed 2022-07-05].
- [28] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, “Composition-based multi-relational graph convolutional networks,” in *International Conference on Learning Representations*, 2020. [Online]. Available: [https://openreview.net/forum?id=ByIA\\_C4tPr](https://openreview.net/forum?id=ByIA_C4tPr)
- [29] D. Vrandečić and M. Krötzsch, “Wikidata: a free collaborative knowledge-base,” *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014, (Accessed 2022-11-03).
- [30] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge,” ser. SIGMOD ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1247–1250.
- [31] T. Kawamura, S. Egami, K. Tamura, Y. Hokazono, T. Ugai, Y. Koyanagi, F. Nishino, S. Okajima, K. Murakami, K. Takamatsu, A. Sugiyama, S. Shiramatsu, X. Zhang, and K. Kozaki, “Report on the first knowledge graph reasoning challenge 2018,” in *Semantic Technology*, X. Wang, F. A. Lisi, G. Xiao, and E. Boteova, Eds. Cham: Springer International Publishing, 2020, pp. 18–34.
- [32] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [33] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [34] M. Kurokawa, “Explainable knowledge reasoning framework using multiple knowledge graph embedding,” in *The 10th International Joint Conference on Knowledge Graphs*, ser. IJCKG’21. Association for Computing Machinery, 2022, p. 172–176.
- [35] T. Ugai, Y. Koyanagi, and F. Nishino, “A logical approach to criminal case investigation,” in *The 1st International Workshop on Knowledge Graph Reasoning for Explainable Artificial Intelligence (KGR4XAI)*, 2021, pp. 1–7. [Online]. Available: [https://kgr4xai.ikgrc.org/papers/KGR4XAI\\_2021\\_paper\\_1\\_cr.pdf](https://kgr4xai.ikgrc.org/papers/KGR4XAI_2021_paper_1_cr.pdf)
- [36] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” in *The Semantic Web*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735.
- [37] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum, “KORE: Keyphrase Overlap Relatedness for Entity Disambiguation,” in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ser. CIKM ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 545–554.
- [38] M. D. Lee, B. Pincombe, and M. Welsh, “An empirical evaluation of models of text document similarity,” in *Proceedings of the annual meeting of the cognitive science society*, vol. 27, no. 27, 2005.
- [39] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96. AAAI Press, 1996, p. 226–231.



knowledge graphs.

**SHUSAKU EGAMI** received his PhD degree in engineering from The University of Electro-Communications, Tokyo, Japan, in 2019. He is currently a senior researcher at the National Institute of Advanced Industrial Science and Technology, Japan. He is also a part-time lecturer at Hosei University, Tokyo, Japan, and a collaborative associate professor at The University of Electro-Communications, Tokyo, Japan. His research interests include the semantic web, ontologies, and



**TAKANORI UGAI** received his PhD degree in engineering from Tokyo Institute of Technology, Tokyo, Japan in 2013. He has been a researcher at Fujitsu Limited, Japan since 1992. He is also a researcher at the National Institute of Advanced Industrial Science and Technology, Japan and a lecturer at Tsukuba University, Ibaraki, Japan. His research interests include knowledge graphs, graph neural networks, and requirements engineering.



**MASATERU OOTA** received a Master's degree in the Department of Mathematics from Hokkaido University, Japan, in 2015. He is currently enrolled in the Doctoral Program in Informatics at the Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan, from 2023. His research interests include knowledge graphs and scene graph generation.



**KYOUNMOTO MATSUSHITA** received his Master's degree in computer science from Kyushu Institute of Technology, Japan, in 2018. He is currently a researcher at Fujitsu Limited, Japan.



**KEN FUKUDA** received his Ph.D. in Information Science from the University of Tokyo in 2001 and was awarded a Research Fellowship for Young Scientists as a doctoral student. He joined the National Institute of Advanced Industrial Science and Technology (AIST) as a research scientist in 2001. He was a visiting lecturer at the University of Tokyo, as well as a visiting associate professor at Waseda University. Since then, he has led multiple national and international projects, covering a broad range of interdisciplinary research from life science to service science. He is currently leading the Data Knowledge Integration Team at the Artificial Intelligence Research Center at AIST. His research interests include knowledge representation, knowledge graphs, data-knowledge integration, and Human-Robot-Interaction.



**TAKAHIRO KAWAMURA** is the Deputy Director of the Research Center for Agricultural Information Technology, National Agriculture and Food Research Organization, Japan. He received M.S. and Ph.D. degrees in electrical engineering from Waseda University. From 2001 to 2002 he was a Visiting Scholar at Robotics Institute, Carnegie Mellon University. His current research interests include knowledge graph construction and its analysis combining semantic technology and machine learning. He was a board member of the Japanese Society for Artificial Intelligence (JSAl) and served as a chair of SIG on Semantic Web and Ontology in JSAl and a member of the Program/Organizing Committees of numerous conferences and workshops including the International Semantic Web conference.