Generalized Property Graphs(GPG) of Property-Value Structures for Knowledge Model

Zhanfan Zhao, Byung-Hun Jeon, Ju-Ri Kim

Abstract: As a popular emerging technology, knowledge graphs (KGs) have become a platform of Web-based knowledge applications and services. Two dominant graph-based knowledge models, RDF and LPG, are widely used to construct large-scale KGs. It is argued that these models have some limitations to cope with complicated knowledge structures. This paper proposes a noble generalized property graph model that can seamlessly realize knowledge structures with compact expressiveness and robust formalism. Since the proposed graph model is compatible with RDF and LPG, it can be practically applied to KGs with effective performance.

Keywords : property graph, knowledge graph, attribute-value matrices.

I. INTRODUCTION

The Web becomes the global application platform for producing, delivering, sharing and consuming information and services. The Web platform provides scalability, security, and reliability for those systems that embrace its simple tenets. So many loosely-coupled, distributed application systems have been implemented on top of the Web platform. Especially, a large-scale, distributed knowledge bases such as Linked Open Data (LOD) have grown to play an essential role on the Web to realize knowledge Graphs(KGs). Several comprehensive KGs such as DBpedia, YAGO, Freebase, and Wikidata containing a large number of plentiful datasets have emerged [1]-[3]. However, despite huge efforts spent to build large-scale KGs, many major problems consistently remain. First of all, knowledge modeling that can provide semantic interoperability and efficient knowledge sharing on the Web should be investigated.

At the moment, Resource Description Framework (RDF) is a prominent model to implement KGs. RDF is a standard graph data model for publishing and interchange data in the Web. The RDF model has the well-defined mechanisms enough to conveniently support the common representation and interoperable semantics of the resources on the Web. Knowledge can be simply represented as subject-predicate-object (SPO) triples. Although RDF is standard for Web-based knowledge modeling and provides the foundation of LOD, RDF modeling has been widely criticized for its awkward structures and semantic interpretations. Especially, the blank nodes and the reification provoke the serious difficulties in querying and searching [4], [9].

On the other hand, another graph data model called Labeled Property Graph (LPG) is prevalent in NoSQL databases. LPG model owns distinctive features, using any number of the key-value pairs to describe the properties of vertices and edges, which makes property graph more expressive and easy to understand for the human being. LPG model shows stronger expressiveness than RDF [8]. Besides, databases based on LPG provide good performance in query and storage for graph data. However, LPG has no concept of global resource identification (that is, no IRIs) or formal semantics and entailment.

This paper proposes a generalized property graph model that can harmonize the distinguishable features of RDF and LPG. The paper formally defines the core conceptual elements to represent knowledge on the Web and presents a generalized property graph data model that embraces the capability of both RDF and LPG. The paper explores the unsolved vital issues in knowledge representation such as N-ary relations, temporal-spatial relations, and reifications.

The remainder of this paper is structured as follows. Section 2 reviews the related work. Section 3 presents the conceptualization of Knowledge and the definitions of the core knowledge elements. In Section 4, a generalized property graph model is presented with its unique features. Section 5 demonstrates the knowledge representation of KGs of the generalized property graph model. Section 6 summarizes the contributions and puts forth the prospects for further work.

II. RELATED WORK

Since graph-based formalisms provide an intuitive and effective way for knowledge modeling, a wide variety of graph-based approaches has been proposed to represent knowledge structures more effectively and practically. In the field of artificial intelligence, several graph-based formalisms such as Semantic Networks and Conceptual Graphs have been developed and a wide range of applications of graph-based methods such as the Entity-Relation model, UML, Topic Maps in the domain of database and information systems development [1]. In recent years a number of research into

Revised Manuscript Received on September 25, 2019

Zhanfan Zhao, College of Information Engineering, Hebei GEO University, Hebei, China. Email: zzfsjz@gmail.com

Byung-Hun Jeon, College of Korean Medicine, WonKwang University, Iksan, Korea. Email: omdjbh@wku.ac.kr

Ju-Ri, Kim*, College of Convergence & Liberal Arts, WonKwang University, Iksan, Korea. Email: cyanic@wku.ac.kr

DOI:10.35940/ijrte.B1087.0982S1019

KGs have been conducted to realize Web-scale KGs containing the large volume of facts. DBpedia, YAGO, LOD Cloud, Freebase, Wikidata, and Cyc are a typical example of the well-known KGs [2], [3]. To implement KGs, two dominant graph data models, RDF and LPG, are widely applied for the materialization of Web-scale KGs [4], [5]. Although RDF and LPG both pursue graph-based knowledge modeling, they adopt different strategies to model knowledge representation.

RDF is a standard data model proposed by W3C for opening, sharing and interchanging data on the Web. RDF has been widely used as the core data model for Linked Open Data (LOD) that enables the Web of Data. Several comprehensive LOD products containing plentiful RDF datasets have emerged such as DBpedia and YAGO [6], [7]. Besides, a large number of industry RDF datasets have been published by numerous researchers, institutions, and companies, which powerfully contribute to the knowledge sharing on the Web. However, RDF modeling has been widely criticized for its awkward structures and semantic interpretations. Especially, the blank nodes and the reification provoke the serious difficulties in querying and searching [8]-[10]. Although RDF reification has been withdrawn from the normative sections in the latest RDF Recommendation [11], the expressive capabilities of RDF remain an unresolved problem [9], [10].

Recently, LPG model emerged from NoSQL database paradigm has received significant attention on account of the good performance in dealing with a huge amount of diverse data generated on the Internet [4], [5], [12]. LPG model owns distinctive features, using any number of key-value pairs to describe the semantic properties of knowledge. LPG provides a more compact, expressive capability to represent graph-based knowledge structures and efficiently store the key-value pairs with index-free adjacency that can allow for good performance in query processing. However, LPG has also revealed some drawbacks although many enhanced functionalities have been proposed for LPG. The conventional LPG model shows a lack of semantic expressiveness to realize semantic interoperability of the knowledge in the open and shared environment of the Web. The value types of the properties used in LPG model are also restricted so that it cannot represent knowledge properties having the compound or structured values.

Since RDF and LPG are dominant graph modeling that has similar objectives, some studies have been accomplished to map from one model to another [4], [13]. However, it is more reasonable to harmonize the distinguishable features of two graph formalisms in order to implement robust KGs effectively.

III. R2RS: SCHEMA-BASED MAPPING OF RDB TO RDF

A large diversity of knowledge modeling approaches has been proposed in many research fields. In recent years, in order to provide knowledge sharing and reuse across heterogeneous domains on the Web, ontology-based approach that provides semantic interoperability has been popular in knowledge modeling. An ontology is a formal, explicit specification of a shared conceptualization of a certain domain [14]. An ontology represents knowledge as a taxonomy of concepts with their attributes, relations, and functions, as well as various domain rules and constraints. Since several ontological conceptualizations of the same domain are possible, the ontological commitment is vital to facilitate the sharing and reuse of knowledge across different domains.

Conceptualization is an abstract, concrete view of a particular domain. In general, through the conceptualization, knowledge of a specific domain consists of the set of the entities and relationships. So knowledge modeling starts with a fundamental distinction between entities and relations.

A. Features of Entities

The entity is the primary element of knowledge structure. An entity is a semantic object encapsulated with the properties or attributes which describe the characteristics of the concept. (In different knowledge modeling approaches, many similar terminologies such as feature, attribute, slot, role, key, and property are used to describe the characteristics of the concept. Especially, the term, property, is used to denote a binary relation in RDF and OWL. This paper uses the term, property, to represent the primitive and contextual meaning of the concept.) So, an entity is generally represented with a set of properties, called property list, in knowledge modeling. The description of the entity using the property list can be found in various areas such as feature-based grammars, frame theory, graph databases, and ontology engineering [15]-[17]. Most of knowledge modeling approaches demonstrate that the property-based conceptualization of the entity is the convenient way to express the diverse aspects of knowledge.

[Definition: Entity] Entity E as a primitive conceptual element of a domain is an atomic building block of knowledge, consisted of a property list that specifies semantic constituents of the entity. (The definition of property list will be given at the later of this section.)

An entity is real-world object (i.e., person, library, and thing) and abstract concept (i.e., culture, technology, and language) inherent in the knowledge domain. An entity can be formalized by property list that is a bundle of primitive conceptual features, as shown in “Fig. 1”. Any properties for the entity can be flexibly appended to represent more informative features and heighten the level of conceptualization without any additional transformations. The possibility to deduce a specific entity employing only property list as shown in “Fig. 1” can underpin the competence of the property-based conceptualization of an entity. The concept Thing#86 in “Fig. 1” can be easily recognized that it represents the specific entity Unicorn. In other words, an entity is created by the property list and cannot be defined without the property list.

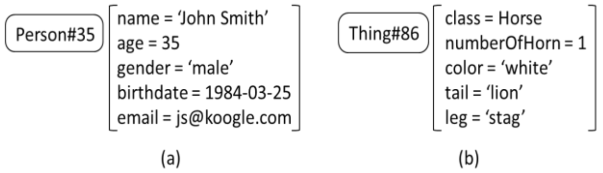

Fig. 1. ASTESJ Logo

B. Features of Relationships

A relationship specifies the semantic association between entities to construct knowledge structure. However, there has been a number of different perspectives to perceive the association between entities form case roles to assertions, and more relations have complicated semantic functions [16], [18], [21]. In reality, relations play a crucial role in representing knowledge structure.

Relationships are also regarded as abstract concepts that reveal similar feature structure to the entity [18]-[20] In additions, since relations, unlike entities holding intensional properties, retain interdependent structural and semantic features, the specification of relationships should be based on some coherent, underlying rationale to represent not only how entities are connected but also how connections are defined with these features. In this paper, relations are defined as follows:

[Definition: Relationship] Relationship R as a semantic association between entities is a binary construct $R(E_i, E_j)$, where E_i and E_j are different entities.

Relations are usually represented as directional links between two connected entities. So the elementary knowledge constructed by a single relation $R(E_1, E_j)$ is usually represented by >, called triple.

In some knowledge modeling, a relation may have three or more entities. The number of entities is called arity or valence of a relation. (A relation R is reflexive if it relates an entity to itself, i.e., E1 = Ej. This paper does not consider reflexivity as other knowledge graphs.) Many studies have proposed the decomposition of relations with arity higher than two, i.e., N-ary relations, into binary relations [22]-[24]. Although some theoretical issues have still remained in the decomposition of N-ary relations, binary relations are common in knowledge modeling [22]. This paper addresses the representation of N-ary relations by means of the property list.

Since modeling relations reflect a worldview of the domain of how entities are related, relations reveal a variety of features depending on the aspects of the associations between entities and the background situations or contexts of knowledge. Indeed, in comparison with entities, relations play a pivotal role in expressing situations (events and states) of knowledge. Relations usually have not only intrinsic features but also these structural and functional features. The variety of types of relations have been analyzed and categorized their features from the diverse perspectives in computational linguistics, AI and cognitive psychology [22], [25]. This paper categorizes the features of relations into 4 major types as follows:

- **Structural features :** Relations as a sort of functions have arity and cardinality. Relations can have arbitrary arity, i.e., unary, binary and N-ary, but only binary relations are allowed in most knowledge graph modeling [22], [25]. Cardinality that specifies the number of related concepts, i.e., 1:1, 1:M, and M:N, are primary constraints of the relations.
- **Functional features:** Binary relations can be characterized by functional features such as reflexive, symmetric and transitive. In OWL, several functional features can be specified with the defined ontologies such as Transitive Property, Symmetric Property, Functional Property, and equivalent Property. (Note that OWL calls relations properties.) These features play a key role in the logical inference of knowledge.
- **Correlative features:** Relations do not exist only between entities. Relations also can be occurred between relations, typically as shown in cause-effect, truth and validity, relations and reifications [22]. This correlative feature between relations is crucial to recognize the background context and situation of knowledge. However, since the correlative feature is expressed implicitly or semantically, it is difficult to capture and represent explicitly.
- **Contextual features:** It is well known that relations, usually represented by the predicates of knowledge expressions, can have temporal (e.g., "since 2015") and spatial (e.g., "in Berlin") information or constraints. This feature generally results from the dynamic behavior and the event characteristic of relations [21], [26]. This information plays a crucial role in knowledge discovery and reasoning. A number of approaches such as N-ary interpretation and reification-based representation have been proposed to explore these features in dynamically changing knowledge [27]. Especially, for temporal information, most approaches regard this information as the same as an entity although it is contextual information depended on the relationships. In addition to temporal and spatial information, much other contextual information such as instrument (e.g., "by the key"), manner (e.g., "3 times a day") and quantity (e.g., "for $300") are also dependent information to relations. A considerable amount of research has been carried out on the underlying structure of relations and emphasized the modalities such as time, manner, instrument, source, and goal as the intrinsic features of relations [16], [20]. It seems quite rational to accept contextual information as the materialization of intensional features of relationships.

The property-based approach to materialize the conceptualization is also appropriate to the representation of relations since its property-value pairs can realize feature structures of relations with simple and efficient manner. "Fig. 2" is a simple example of property-based modeling of relations. (More examples with complicated relationships will be discussed in Section 4.)

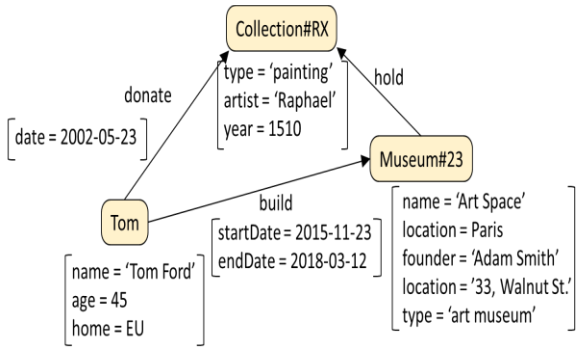

Fig. 2. Entities and Relations Description of Property-based modeling

C. Features of Relationships

Semantic constituents inherent to the concepts can be decomposed into property structure. Properties (sometimes called slots, attributes, frames, or roles) are the intensional characteristics of an entity and relation. However, the notion of the property is quite ambiguous due to the diverse perspectives in different realms. This paper defines the property as follows:

[Definition: Property] A property P is a unary function P(X) to represent intensional features of the entity and relation, where X is either an entity or a relation. The property value can be literal, integer, date, and identifiable data.

The property is special unary relation where the range of a relation is the value of an identifiable data type. (Note that "property" is used to denote a binary relation in RDF(S) and OWL.) Properties are intrinsic characteristics, for examples, age(John)=33, capital(German)=Berlin, and height(Everest)=8848. The adjunctual meanings also can be properties, for examples, mother(Tom)=Mary, birth date (Susan)=2003-05-25, and nickname(Hawaii)='The Aloha State'.

In additions, the contextual and functional features of relations are represented by means of the property structure. For example, the temporal and special property of relation M=marry(David, Jane) can be specified as date(M)=1988-04-17 and place(M)=Paris over M.

The definition of the property can be applied to judge whether a conceptual element is a property or relation. However, even though isA(John)=Professor is a unary relation, isA is not a property to represent intrinsic meaning, but subsumption relation between concept classes. It is worth noting that the binary functionality of isA can be observed when attaching some properties to isA, for example, since (isA(John, Professor))=1988.

IV. FORMAL DESCRIPTION OF GENERALIZED PROPERTY GRAPH

This section describes the definition of a noble knowledge representation method called generalized property graph. The characteristics of the property list that is the core structure of the generalized property graph are also described.

A. Definition of Generalized Property Graphs

Although the generalized property graph (GPG) model adopts the approaches of typed feature-structured formalisms and LPG modeling, GPG embraces the open-world features of RDF and NoSQL conceptions so that it can realize Web-scale KGs. GPG provides powerful expressiveness of knowledge structures and efficient store of property lists with index-free adjacency that can allow for fast querying.

Definition (Generalized Property Graph): A GPG G is a directed, labeled, attributed, multi-relational graph consisting of $G=(E,R,K,P,L,\alpha,\beta)$, where E is a set of entities, R is a set of relations, P is a set of property lists, L is a set of labels, $\theta:(E\cup R)\longrightarrow L$ is a label assignment, and $\alpha:(E\times P)\longrightarrow E$ and $\beta:(R\times P)\longrightarrow R$ is a conceptual function for entities and relations, respectively.

In GPG, entities are represented as nodes and relations as edges. Both nodes and edges are labeled with their semantic functions. The edges are directed and can be multiple-edges between any two nodes.

Entities denoted by nodes have a property list representing conceptual attributes inherent in the entities. Entities can have one or more labels. Entity labels can play a vital part in specifying ontological classes of entities. This makes it possible to form conceptual schema or hierarchy of ontological classes efficiently.

Relations denoted by edges have a direction to connect two entities. As the mandatory feature of GPG model, similar to LPG, every relation must have one and only one label to represent the edge uniquely. Much like entities, relations also can have their property list. The property list of relations usually describe the eventual or contextual attributes such as time, location and modality when the relationship is built between two entities.

Properties vs. Relations: Properties are special relationships where the ranges of a relationship are values of a data type (e.g., dates, age) as opposed to instances of a concept.

Both entities and relations have labels, however, their applications are different. The entity labels are a way to assign ontological classes to categorize vertices by means of their semantic features. The entity labels are similar to rdf:type of RDF, but more efficient and powerful. The entity labels can be used for various purposes such as sub-graph creation, efficient GPG data store, and ontology schema generation. In other word, labels can be used to assign types to entities. On the other hand, every edge has mandatorily one and only one label that represents the relationship between two connected entities. The edge label plays a role of the unique identifier of relationship.

Although GPG is conceptually similar to the conventional LPG, it has its distinct features that are more uniform and powerful capability based on the property lists. These features can seamlessly embrace the idiosyncratic functions of both RDF and LPG, and GPG can provide consistent modeling for entities and relations. "Fig. 3" is a typical example of GPG.

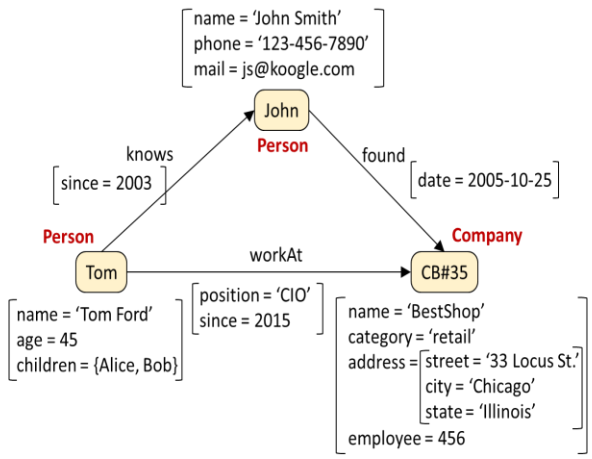

Fig. 1. Example of Generalized Property Graph

B. Characteristics of Property Lists

The property list is the foundational mechanism of GPG to describe the attributes of both entities and relationships. Since the attributes are the intensional characteristics of an entity, object or relation, the property usually represents intrinsic or conceptual features such as color, weight, and size for entities, time and location for relations. The metadata or ontological vocabularies can be used for the property. Since the property list is essential for expressing non-relational data, it should be distinguished from relationships between entities that are generally represented by the edges.

■ Types of the Properties

As the most convincing way of knowledge modeling, the diverse types of property-oriented approaches can be easily encountered in many research fields. Much well-known formalism such as feature structures and attribute value matrices (AVM) in computational linguistics, the key-value pairs in NoSQL databases, slot-and filler of frame theory in artificial intelligence are shared the fundamental notions and similar representation structures to model knowledge [3], [5], [16], [23]. However, since they use different approaches to formalize the properties from their perspectives, it is required to generalize the aspects of the property from the conceptual point of view.

The popular KG modeling such as RDF(S), OWL and LPG provide a more general mechanism to define the properties efficiently. (Note that the notions about the property in this paper are different from RDF(S) and OWL.) However, the distinction between the property and relationship is ambiguous and the lack of explicit, semantic consideration. Since the property mechanism is the semantic function of knowledge representation, the categorization based on the type of property values, datatype property and object property, is not appropriate. This paper formally defines a property is a unary function to represent intensional features of the entity and relation. The property can semantically be defined into four categories as follows:

- **Attributive property**: properties to represent intensional semantics of the entities. For examples, color(Apple)='red', age(Grandmother)=78 and gender(Paul)='male' are typical attributive properties.
- **Adjunct property**: properties to represent the contextual or situational facts accompanied by relations. The temporal; spatial and modal facts occurred with relations, for examples, since(workAt(John, Google))=2003 and location(meet(John, Bob))='Starbucks', are typical adjunct properties.
- **Constituent property**: properties to relate to semantic roles of the entities. In general, some N-ary relations can be reduced into binary relations. When diminished the degree of arity, some entities that are strongly coupled with the relation become the values of the constituent property. For example, a typical 3-ary relation 'John gave Mary iPad.' Can be converted into 'John gave iPad to Mary.' In this case, a property related to the entity Mary will be a constituent property. The constituent property is effective to handle N-ary and other complex relationships between entities. Since the constituent property can be described with appropriate ontologies, the structural and semantic expressiveness will not be lost.
- **Correlative property**: properties to specify the contextual association between facts. As relations specify the factual association between entities, the representation that expresses the contextual relationships between facts are indispensable to model complex knowledge structures. This requests the reification of the facts using meta-knowledge such as provenance, trust, and certainty [10]. (Although RDF reification has been withdrawn from the normative sections in the latest RDF Recommendation [11], the reification representing triples about triples is a basic requirement for the RDF data model but remains an unresolved problem.) Although a considerable amount of research has been proposed, most of them have focused on structural consistency of knowledge representation. In fact, the reified statement is an annotated or restricted fact by other controlling entity and retains the locality characteristics between relations. So, the reification can be modeled by means of correlative property between relations since relations are the binary association between entities by definition. For example, some correlative properties, assert and asserted, can insert into property lists as assert(believes(John, Galileo))=found and asserted(found(Galileo, Saturn))=believes for 'John believes that Galileo found Saturn.'.

The property list provides a convenient and efficient way to tag or annotate the intensional semantics as well as the contextual or situational information for entities and relations. The expressiveness of the property list can resolve the problematic issues of knowledge modeling such as N-ary and reification.

■ Extended Property Structures

The hierarchical property structures have been widely used in feature-based systems to provide more concise and understandable conceptualization for compound property [15], [23].

The hierarchical property structures can localize the structured data of the properties and provide a preferable conceptualization of entities and relationships. As discussed in feature-based systems, the hierarchical property structures also have complete theoretic basis and application use cases.

The effectiveness of hierarchical property structures can be observed in Figure 3 and 4. The hierarchical property structures provide a compact and concise representation of structured knowledge.

V. KNOWLEDGE GRAPHS USING GPG

GPG has a powerful modeling capability and shows explicit expressiveness. So GPG can tackle not only the cumbersome problems of RDF and LPG but also the general issues related to knowledge representation. The following is a typical example that GPG addresses the reification and blank nodes of RDF. As shown in “Fig. 4”, GPG localizes the relations of the reification even though it has additional information. The hierarchical property value structure provides an efficient way to remove awkward blank nodes. Moreover, GPG is facile to employ structured data types such as sets or lists.

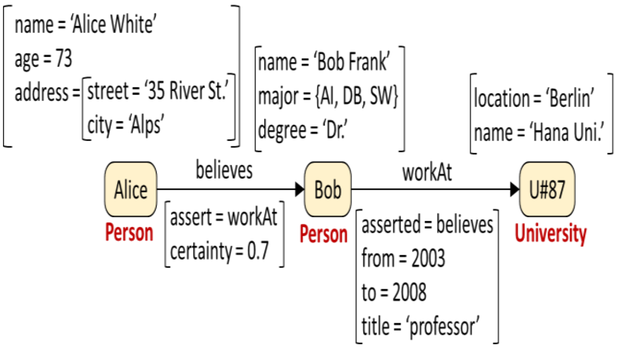

Fig. 4. GPG representation of complex knowledge structure

VI. CONCLUSION

In recent years a number of sizable KGs have been developed, the largest ones containing more than 100 billion facts. However, to disseminate KGs, many practical issues with knowledge expressiveness, completeness, and performance should be solved. The efficient knowledge modeling is essential to realize KGs and .share them on the Web. This paper addresses a new knowledge modeling approach to resolve the cumbersome problems that are significant to represent complicated knowledge structures.

There are two dominant graph-based knowledge models, RDF and LPG, popular for KGs. Although these graph models have plentiful knowledge modeling capabilities, they reveal critical limitations on the description of important knowledge structures. This paper proposes a generalized property graph model from the perspective of conceptual modeling of knowledge. This paper explores the core conceptual elements for knowledge modeling and defines their functions. The property list with enhanced capabilities is proposed to represent complicated knowledge structures more efficiently. The generalize property graph provides compact and comprehensive knowledge modeling and solves some intractable problems such as N-ary relations and reifications. Moreover, since the proposed generalized property graph is compatible with RDF and LPG, it can be efficiently applied to transform or federate the current well-known KGs. The sound implementation will be investigated as a further research project.

ACKNOWLEDGMENT

This research was supported by the MISP(Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW)(2018-0-01880) supervised by the IITP(Institute of Information & communications Technology Planing & Evaluation) (2018-0-01880).

REFERENCES

1. Pavlić, Mile, Ana Meštrović, and Alen Jakupović, “Graph-based formalisms for knowledge representation.” Proceedings of the 17th world multi-conference on systemics cybernetics and informatics (WMSCI 2013), 17th ed. vol. 2, 2013.
2. Ringler, Daniel, and Heiko Paulheim, “One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & co.” Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Springer, Cham, vol. 10505, 2017, pp. 366-372
3. Paulheim, Heiko, “Knowledge graph refinement: A survey of approaches and evaluation methods.” Semantic web, 3rd ed. vol. 8, 2017, pp. 489-508.
4. Das, Souripriya, et al, “A Tale of Two Graphs: Property Graphs as RDF in Oracle.” EDBT, vol. 82, 2014, pp. 762-773.
5. Margitus, Michael, Gregory Tauer, and Moises Sudit, “RDF versus attributed graphs: The war for the best graph representation.” 2015 18th International Conference on Information Fusion Washington, DC, IEEE, 18th, 2015, pp. 200-206.
6. Hoffart, Johannes, et al, “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia.” Artificial Intelligence, vol. 194, 2013, pp. 28-61.
7. Heath, Tom, and Christian Bizer, “Linked data: Evolving the web into a global data space.” Synthesis lectures on the semantic web: theory and technology, 1th ed. vol. 1, 2011, pp. 1-136.
8. Kanmani, A. Clara, T. Chockalingam, and N. Guruprasad, “RDF data model and its multi reification approaches: A comprehensive comparitive analysis.” 2016 International Conference on Inventive Computation Technologies (ICICT), IEEE, vol. 1, 2016, pp. 1-5.
9. Hernández, Daniel, Aidan Hogan, and Markus Krötzsch, “Reifying RDF: What works well with wikidata?.” Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems, CEUR Workshop Proceedings, vol. 1457, 2015, pp. 32-47.
10. Hartig, Olaf, and Bryan Thompson. Foundations of an alternative approach to reification in RDF. arXiv preprint arXiv:1406.3399, 2014.
11. W3C, RDF 1.1 concepts and abstract syntax. W3C Recommendation 25 February 2014, https://www.w3.org/TR/rdf11-concepts/.
12. Rodriguez, Marko A., and Peter Neubauer, “Constructions from dots and lines.” Bulletin of the American Society for Information Science and Technology, 36th ed. vol. 6, 2010, pp. 35-41.
13. Hartig, Olaf, “Reconciliation of RDF* and property graphs.” arXiv preprint arXiv:1409.3288, 2014, pp. 1-18.
14. Studer, Rudi, V. Richard Benjamins, and Dieter Fensel, “Knowledge engineering: principles and methods.” Data & knowledge engineering, Issues 1-2, vol. 25, 1998, pp. 161-197.
15. Carpenter, Bob. The logic of typed feature structures: with applications to unification grammars. logic programs and constraint resolution. Cambridge University Press, 2005, pp. 32.
16. Rouces, Jacobo, Gerard de Melo, and Katja Hose, “FrameBase: Enabling integration of heterogeneous knowledge.” Semantic Web, 6 ed. vol. 8, 2017, pp. 817-850.
17. Levine, Robert D., and Walt Detmar Meurers, “Head-Driven Phrase Structure Grammar: Linguistic approach, formal foundations, and computational realization.”

Encyclopedia of language and linguistics, vol. 2, 2006, pp. 1-24.

18. Bonnotte, Isabelle, “The role of semantic features in verb processing.” Journal of Psycholinguistic Research, 3th ed. vol. 37, 2008, pp. 199-217.
19. Ma, Minhua, and Paul Mc Kevitt, “Visual semantics and ontology of eventive verbs.” International Conference on Natural Language Processing. Springer, Berlin, Heidelberg, vol. 3248, 2004, pp. 187-196.
20. Fellbaum, Christiane. WordNet. The Encyclopedia of Applied Linguistics. 2012.
21. Bergman, Michael K. A Knowledge Representation Practionary: Guidelines Based on Charles Sanders Peirce. Springer. 2018.
22. Khoo, Christopher SG, and Jin-Cheon Na, “Semantic relations in information science.” Annual review of information science and technology, 1th ed. vol. 40, 2006, pp. 157-228.
23. Véronis, Jean, and Nancy Ide, “A feature-based model for lexical databases.” Proceedings of the 14th conference on Computational linguistics, Association for Computational Linguistics, vol. 2, 1992, pp. 588-594.
24. Chein, Michel, and Marie-Laure Mugnier. Graph-based knowledge representation: computational foundations of conceptual graphs. Springer Science & Business Media. 2008.
25. Sowa, John F, “Conceptual graphs.” Foundations of Artificial Intelligence, vol. 3, 2008, pp. 213-237.
26. Schuler, Karin Kipper. VerbNet: A broad-coverage, comprehensive verb lexicon. University of Pennsylvania. 2005.
27. Rula, Anisa, et al, “On the diversity and availability of temporal information in linked open data. International Semantic Web Conference.” Springer, Berlin, Heidelberg, vol. 7649, 2012, pp. 492-507.

AUTHORS PROFILE

Zhanfang Zhao is an assistant professor of Hebei GEO University, Hebei, China. She attended Hebei Institute of Technology in 2000 and University of Mining & Technology, Beijing, China in 2005. She is a senior researcher of Smart Technology Lab in WonKwang University, Korea. She received her PhD degree in Department of Computer Engineering, WonKwang University in 2019. Her main research areas are artificial intelligence, ontology, knowledge management and NoSQl databases

Byung-Hun Jeon is currently professor for Department of Pathology, College of Korean Medicine and Chairman, Korean Society of Korean Medicinal Pathology, also served as editorial manager of Korean Journal of Korean Medicinal Physiology and Physiology. PhD & Master in Department of Korean Medicine, Graduate School of WonKwang University, Republic of Korea. MD graduated from College of Korean Medicine, Wonkwang University on 1986. Dr. Jeon has over 29years experiences in education of pathology and study in the field of anti-cancer herbal drug screening and is the author of more than 150 scientific publications in this field. He has been to NHLBI, NIH in USA for 2 years on 1998~2000 as a visiting scientist

Ju-Ri Kim is an assistant professor of Convergence & Liberal Arts College of WonKwang University. She received her PhD degree in Department of Computer Engineering, WonKwang University in 2009. She is a steering committee member of many academic societies, to name a few, Korea Convergence Society and Korea Computer Education Society. She has participated in a reviewer of many national research projects. She is interested in 2D /3D image processing, sematic annotation of multimedia resources, Linked Data and machine learning.