Journal of Theoretical and Applied Information Technology
31st October 2019. Vol.97. No 20
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTATION OF LINK POLICY CONSULTING IN-DEPTH SEARCHING SYSTEM FOR LINKED OPEN DATA CLOUD

[1]YONGLAK SOHN

[1] Professor, Dept. of Computer Engineering, Seokyeong University, Seoul, Korea

E-mail: [1]syl@skuniv.ac.kr

ABSTRACT

For Linked Open Data (LOD) cloud, an implementation of semantic web, this research suggested the effectiveness of link policies in order to evaluate sameness of entities in different LODs. Operating the link policies reduced heavy burdens of explicitly specifying sameAs links between the entities. It overcame the problem of omitting entities which must have been searched properly from LOD cloud. Knowledge expansion is LOD cloud's major goal. Until now, it has been realized by the sameAs links. The links have been specified explicitly and appended to LODs periodically. At the time of searching, therefore, considerable number of entities had been excluded from being searched. In addition, the burden of specifying sameAs links resulted in severe paucity of linkages in LOD cloud and knowledge expansion has been limited consequently. To overcome the limitations, this research suggested LODs to prepare their own link policies. For the policy establishment, this paper devised vocabularies set. Instead of following the sameAs links which have been specified explicitly and appended to LODs, searching process consults link policies of source and target LODs and evaluates the sameness between source and target entities. As the link policies substitute the sameAs links, burdens of specifying and maintaining the sameAs links have been reduced. By consulting the policies at the time of searching, moreover, any changes in LOD cloud will not be omitted from the searching results. In order to verify the effectiveness of link policies, this research implemented In-Depth Searching (IDS) system. For a given searching request, IDS progresses in depth by consulting the link policies and by accessing LODs. Analysis on the effectiveness has been performed from the standpoint of recall ratio and precision ratio.

Keywords: *Linked Open Data Cloud, SameAs Link, Link Policy, Ontology, Semantic Web*

1. INTRODUCTION

Currently, web is composed of pages and links among them. However, the pages are lacking in semantics and thus computers have severe restrictions on utilizing the contents within them in detail. Linked Open Data (LOD) cloud, a practical implementation of semantic web, has overcome such restrictions by providing semantic structures, named ontology [1-6]. Since 2007, LOD cloud has been in progress together with W3C's supports and currently has 1,163 LODs as collaborators [7-11]. In LOD, a fact about an entity is presented in accordance with Resource Description Framework (RDF) model which describes the fact with {<subject> <predicate> <object>} and serializes it as RDF triple [12-14].

In Figure 1, two RDF triples in LOD_A describe that Personal:Alice's gender is 'Female' and her homepage's URI is 'http://www.Alice.net'. Predicates foaf:Gender, foaf:Homepage present semantic relationships between subject Personal:Alice and objects 'Female', 'http://www.Alice.net'. Together with providing the semantic structures, the goal of LOD cloud is to provide knowledge expansion properly.

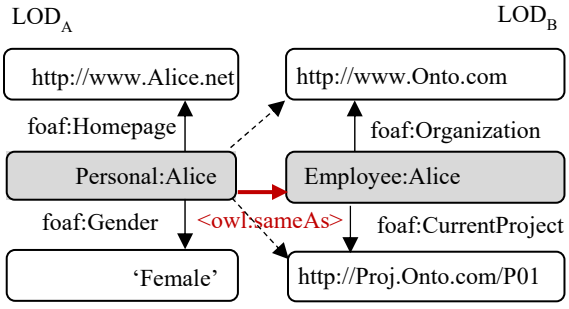

Figure 1: Knowledge Expansion with Identical Links

By asserting that the two subjects, LOD_A's Personal:Alice and LOD_B's Employee:Alice, are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the same by means of OWL's predicate <owl:sameAs>, a knowledge obtained by accessing Personal:Alice expands with the facts that Employee:Alice has worked in a company whose homepage is 'http://www.Onto.com' and currently participated in a project identified as 'http://Proj.Onto.com/P01'. The link between Personal:Alice and Employee:Alice is called sameAs link.

Although the sameAs links perform a key role in knowledge expansion, current LOD cloud suffers from paucity of them. 44% of LODs in LOD cloud do not have any sameAs links to other LODs and thus remain in data silo as ever. 27% of LODs have the links to only one or two LODs [15-18]. Major cause of the paucity is that generating and maintaining large amount of sameAs links in the form of RDF triple have been heavy burden for LOD publishers [19].

To reduce the burdens, previous works [20, 21] proposed methods for automatic sameAs link generation. They generated sameAs links by evaluating objects of source and target candidate RDF triples whose predicates have been chosen for providing their objects. More elaborated works [22-26] considered syntactic features, such as inverse functional property, cardinality, inclusion and so on of RDFS, OWL, and OWL2 to the similarity evaluations of source and target objects. These works had something in common. They generated the sameAs links as RDF triples. To publish the sameAs links into LOD cloud, they cannot help appending the RDF triples to an LOD periodically.

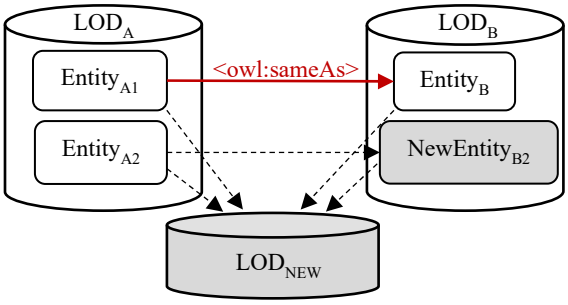

Figure 2: Inelasticity of explicitly specified sameAs links

Periodical appendence of the sameAs links became inelastic to searching requests because changes in an LOD could not appear in searching results until LOD publishers would generate new sameAs links and append them to the LOD. As depicted in Figure 2, the sameAs link has been serialized as an RDF triple, {<Entity$_{A1}$> <owl:sameAs> <Entity$_{B1}$>} and appended to LOD$_A$. It gives a chance that RDF triples whose subjects are <Entity$_{A1}$> in common can be expanded with other RDF triples whose subjects are <Entity$_{B1}$> because <Entity$_{A1}$> and <Entity$_{B1}$> have been asserted to be the same. Notwithstanding that RDF triples whose subjects are <NewEntity$_{B2}$> in common have appeared newly and <NewEntity$_{B2}$> is identical with <Entity$_{A2}$>, however, if LOD$_A$'s publisher did not specify a sameAs link {<Entity$_{A2}$> <owl:sameAs> <NewEntity$_{B2}$>} yet and thus did not append the link triple to LOD$_A$, an access to <Entity$_{A2}$> could not be expanded with RDF triples of <NewEntity$_{B2}$>. Furthermore, in case that LOD$_{NEW}$ has participated to LOD cloud newly, until the whole sameAs links have been specified explicitly and published to LOD cloud together with LOD$_{NEW}$, all the RDF triples of LOD$_{NEW}$ cannot participate to any knowledge expansions.

To overcome the limitations, instead of periodically generating and appending the huge amount of sameAs links, this paper suggests that every LOD establishes its own link policy and publishes it into LOD cloud. Link policy is composed of a number of constraints which have been prepared to evaluate the degree of sameness between source and target entities in different LODs. For specifying the link policy in detail, this research has devised a set of vocabularies. The link policy is consulted at the time of searching in-depth and therefore omitting any modifications in an LOD, which must have been included in searching results, becomes intrinsically impossible. To verify effectiveness of operating the link policies, this research has implemented In-Depth Searching (IDS) system. IDS starts from a surface LOD, which has been selected by a query requester, and gets entities which will be representative subjects of final results. By consulting specifications in link policy elaborately, IDS finds target LODs for next depth searching and target entities within the LOD. IDS repeats these steps until it reaches at a depth level which has been requested in the user. RDF triples searched are reorganized as final results which also have been equipped with their own ontologies customized for further usages.

2. RELATED WORKS

There are two approaches for generating sameAs links in LOD cloud, one is using standard identifiers and the other is using degree of similarity between objects in source and target RDF triples. The approach of using standard identifiers gives attention to <owl:InverseFunctionalProperty>

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which has a feature of inverse function [13]. For example, assume that there are two RDF triples {<Entity$_{A1}$> <Predicate_IFP> <Entity$_{B1}$>}, {<Entity$_{A2}$> <Predicate_IFP> <Entity$_{B1}$>} and <Predicate_IFP> has been specified to have a feature of <owl:InverseFunctionalProperty>. Because two objects are the same, two subject entities, although they have been identified with different URIs, are convinced to be linked with a sameAs link. This approach can be utilized to RDF triples whose objects are standard identifiers such as ISBN(International Standard Book Number), GTIN(Global Trade Item Numbers), or ISIN(International Securities Identification Numbering) and so on. But the limitation, of course, is that this approach can be applied only to the situations of using such standard identifiers [13].

The other approach which utilizes a similarity of objects in source and target RDF triples begins with understanding ontologies of the source and target LODs. Based on semantics of specifications in the ontologies, it chooses pairs of predicates from the ontologies. If it found objects which were connected to the predicate pairs and recognized them to be similar sufficiently, it would establish a sameAs link between their corresponding subject entities. The approach of using standard identifiers can operate successfully on this approach. Researches such as [20-26] have carried out to realize this approach. They have applied weighting strategy for the predicate pairs and paid elaborated considerations for the degree of similarity of source and target objects as long as the objects are literals. In case of [20], it utilizes a predicate of <rdfs:labels> which has a feature of generality. Using the predicate, it then confines target RDF triples to be appropriate candidates which will be sure to provide RDF triples to final results. Based on the predicate pairs selected, it evaluates similarity degrees of source and target objects elaborately and finds out pair of subject entities which will be linked as the same. In case of [21], it focuses on improving performance of [20]. Analogous to [20], it has built up a set of RDF triples whose subject entities are candidates to be linked with the sameAs links. From the candidate set, it selects family samples and finds subject entities which are near sufficiently to a sample by using trigonometric inequality based on the concept of spatial location point. It has accomplished performance improvement by excluding subject entities in advance which are never to be the same.

On the top of [20] and [21], [22-26] have considered syntactic features of subject, predicate, and objects in RDF triples. For syntax observations, [22] aims at RDFS and OWL and [23] adds OWL2. Owing to participation of OWL2, inference features of OWL2 have contributed to the evaluation of sameness between source and target subject entities. [24-26] have focused on evaluating a reliability of sameness between source and target subject entities. Founded on the orthodox information theory, RDF triples in an LOD are assertions of a publisher of the LOD [8, 11]. SameAs links are the assertions as well. Reliability of the sameAs links between subject entities is largely responsible for reliability of their corresponding RDF triples which have been depended on the reliability of the LOD's publisher [16]. For the reliability observations in circumstances of Web, Page Ranking algorithm of Google is a representative work [27]. According to the work, reliability of a page currently searched increases in proportion with the number of pages which referenced the searched page. It also increases according as reliabilities of the pages, which referenced the searched page, have been high. Although there are slight differences, [24, 25] have adopted the approaches of [27].

These works [20 – 26] have contributed to enrich the sameAs links in LOD cloud. As explained in chapter 1, however, because of generating RDF triples for the sameAs links and of appending them to LODs periodically, changes in LOD cloud cannot participate in searching results until the sameAs links appear explicitly in LOD cloud.

3. VOCABULARIES FOR LINK POLICY SPECIFICATION

Link policy suggested by this research is composed of specifications for topic confinement and predicate matching. Topic confinement specifies a topic of entities, which are candidates to link, in order to confine target entities whose topic corresponds with that of source entities. For example, by specifying that a topic <http://ko.dbpedia.org/class/movie> of Korea DBpedia corresponds with a topic <http://fr.dbpedia.org/class/film> of France DBpedia, a searching process is able to confine entities in France DBpedia LOD whose topic is 'film' as long as it looks for entities in order to link them with an entity whose topic is 'movie' in Korea DBpedia.

Predicate matching specifies predicates pair in source and target LOD that have been recognized as semantically the same. By inspecting similarity of source and target objects whose predicates have been matched, we can figure out sameness degree

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Vocabularies for Link Policy Specification

Vocabulary	Role	Subject	Object
lp:linkpolicy	Link policy registration	SourceLOD	BlankNode_A
lp:targetLOD	Target LOD registration	BlankNode_A	Target LOD
lp:regTopicConfine	Topic confinement registration	BlankNode_A	BlankNode_B
lp:sourceTopicPredicate	Predicate for source LOD's topic confinement registration	BlankNode_B	Predicate for source LOD's topic confinement
lp:sourceTopicConfine	Type for source LOD's topic confinement registration	BlankNode_B	Type for source LOD's topic confinement
lp:regTargetConfine	Target LOD's topic confinement registration	BlankNode_B	BlankNode_C
lp:targetTopicPredicate	Predicate for target LOD's topic confinement registration	BlankNode_C	Predicate for target LOD's topic confinement
lp:targetTopicConfine	Type for target LOD's topic confinement registration	BlankNode_C	Type for target LOD's topic confinement
lp:predicateMatching	Predicate matching registration	BlankNode_B	BlankNode_D
lp:sourcePredicate	Source LOD's matching predicate registration	BlankNode_D	Source LOD's predicate
lp:targetPredicate	Target LOD's matching predicate registration	BlankNode_D	Target LOD's predicate

of source and target entities which are subjects in source and target RDF triples. For example, together with topic confinement stated before, let's assume that a predicates pair, <http://ko.dbpedia.org/property/movieDirector> and <http://fr.dbpedia.org/property/filmDirector>, have been specified to be matched because of their semantic sameness. When in-depth searching proceeds from a source entity in Korea DBpedia to a target entity in France DBpedia, let's assume that their topics are the same and their predicates are semantically the same as well. Moreover, if their objects were very similar, the target entity would be sufficient to be a destination of a sameAs link which had started from the source entity in Korea DBpedia. If there were two or more predicate matching specifications, the sameness between the source and target subject entities would be reinforced.

Table 1 presents vocabularies for specifying link policies. They comply with RDF model and thus link policies specified by them can be accessed in the same way as for LODs. Namespace 'lp' corresponds to 'http://linkpolicy.org/ontology'. In fact, link policy is an ontology that specifies various constraints in detail in order to evaluate the sameness of subjects on different LODs for the purpose of linking them as identical ones.

LOD publisher starts link policy specification with lp:linkpolicy. As an object, lp:linkpolicy has BlankNode_A which becomes a starting point of source LOD's policy specifications. lp:targetLOD, a predicate of BlankNode_A, registers URI of target LOD's SPARQL Endpoint. SPARQL Endpoint is a process of an LOD that gets SPARQL query, processes the query in an LOD, and then returns results to the requester [28]. SPARQL is W3C's standard query language for accessing RDF triples [29, 30].

A topic to confine is registered with lp:regTopicConfine and gets BlankNode_B as an object. With lp:sourceTopicPredicate, BlankNode_B specifies predicates which will be applied to the topics confined before. With lp:sourceTopicConfine, BlankNode_B also specifies topics to be confined in SourceLOD. Registration of topics to confine in TargetLOD is specified with lp:regTargetConfine and gets BlankNode_C as an object. With lp:targetTopicPredicate and lp:targetTopicConfine, LOD publisher specifies predicates and topics for confining RDF triples to be searched from TargetLOD elaborately during in-depth searching. These confinements will play an important role in lightening considerable burden on searches in TargetLOD.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

lp:predicateMatching takes on a role of specifying a pair of source and target predicates which are semantically almost or wholly the same. Predicate specified with lp:predicateMatching must meet the topics that have already been specified to confine and therefore its subject is BlankNode_B which has been the object of lp:regTopicConfine. lp:predicateMatching gets BlankNode_D as its object. lp:sourcePredicate and lp:targetPredicate have BlankNode_D as their joint subject. LOD publisher employs them to specify matching pair of predicates in source and target LODs to declare that they are identical in semantics.

4. IMPLEMENTATION OF IN-DEPTH SEARCHING SYSTEM

In-Depth Searching (IDS) system has been implemented in order to make certain that operating link policy is practically useful for knowledge expansion which is LOD cloud's major goal. To deal with SPARQL queries, IDS adopted Apache Jena 3.1.0 API. Figure 3 presents overall system architecture of IDS. IDS serves user's searching requests as follows: surface searching, in-depth searching with link policy consulting, and results reorganizing.

4.1 Surface Searching

User's access to LOD cloud begins with selecting 'Surface LOD'. To 'Surface LOD', user sends a searching request which is composed of {User_Query, SPARQL_Endpoint, Depth_Level, Sameness_Degree}. User prepares 'User_Query' in accordance with SPARQL standard. 'SPARQL_Endpoint' is a URI of SPARQL endpoint of 'Surface LOD'. 'Depth_Level' is an integer of zero or more which declares a depth limit to move forward for searching. If 'Depth_Level' were zero, IDS would perform the search only to 'Surface LOD'. If 'Depth_Level' were greater than zero, IDS would launch in-depth searching to 'Next LOD'. 'Sameness_Degree' is a value of between 0.0 and 1.0. It sets minimum requirement for sameness degree of subject entities in source and target RDF triples. Before evaluating the sameness, topics and predicates of the source and target RDF triples must coincide with link policy of 'Source LOD' in respects of topic confinement and predicate matching.

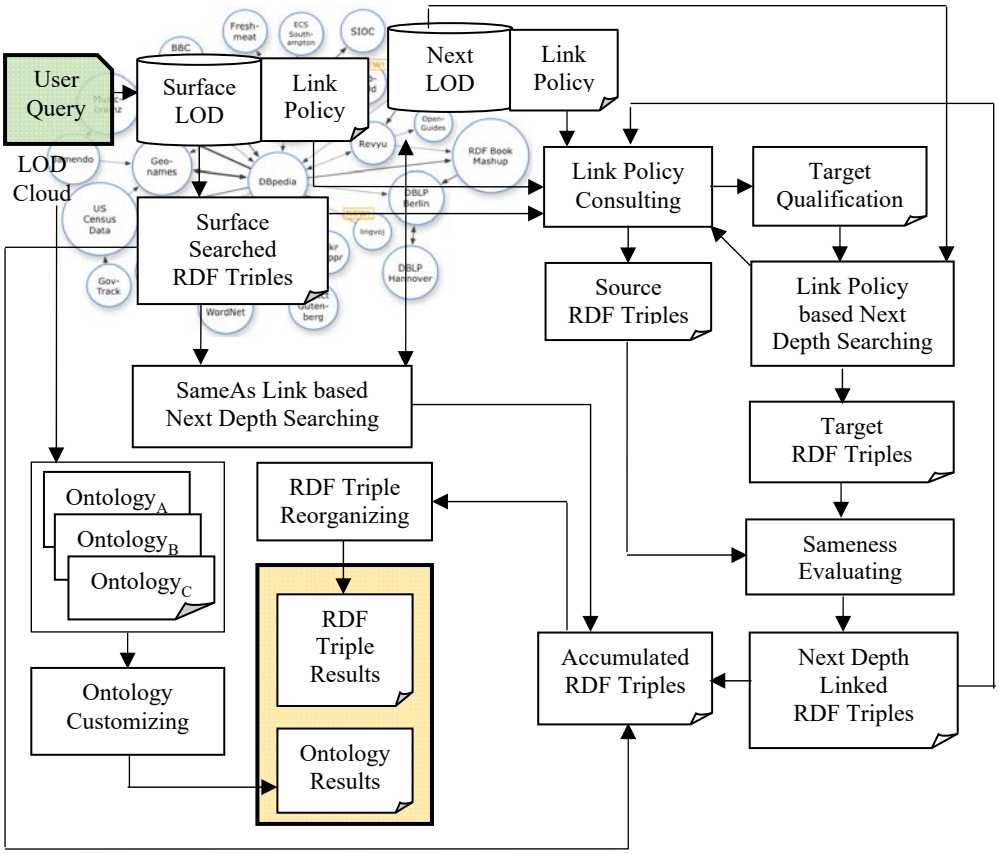

Figure 3: System Architecture of In-Depth Searching System

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

With this prerequisite, similarity degree of objects in source and target RDF triples becomes the sameness degree of their subjects. If user requested 'Sameness_Degree' to be 1.0, IDS would regard the source and target subjects as the same only if their objects of the selected predicates pair were exactly the same. If either or both of the objects were URIs, IDS would decide their similarity as either 0 or 1 only. IDS passes every RDF triple in 'Surface Searched RDF Triples' to 'Accumulated RDF Triples' which at last will join in the searching results because user has selected 'Surface LOD' with his conviction that the LOD is trustworthy sufficiently.

4.2 In-Depth Searching with Link Policy Consulting

IDS would launch in-depth searching if it received 'Depth_Level' of one or more. It performs In-depth searching in two ways: Link policy based next depth searching and SameAs link based next depth searching.

4.2.1 Link policy based next depth searching

IDS searches in-depth by aid of link policies prepared by LODs. From the link policy, IDS finds out next LOD to visit. To progress the first depth from 'Surface LOD' in Figure 3, 'Link Policy Consulting' looks into a 'Link Policy' of 'Surface LOD' and finds out policy specifications about topic confinement and matched predicates pairs. Based on the specifications, it selects RDF triples from 'Surface Searched RDF Triples' which accord with the specifications and makes up 'Source RDF Triples'.

'Link Policy Consulting' finds out target predicate after checking out that the predicate was specified in the link policy as a pair of the predicate in the source RDF triple. It then composes 'Target Qualification' as {Target_LOD, Target_Topic, Target_Predicate} which describes that only RDF triples in 'Target_LOD' whose topic is 'Target_Topic' and predicate is 'Target_Predicate' can be candidates for the sameness evaluation. 'Link Policy based Next Depth Searching' composes SPARQL query together with 'Target_Topic' and 'Target_Predicate' as conditions for searching in 'Next LOD'. It sends the query to SPARQL endpoint of 'Next LOD'.

Searching results from 'Next LOD' are stored in 'Target RDF Triples'. 'Sameness Evaluating' first examines that source and target RDF triples have the same topic. It then evaluates similarity of objects of source and target RDF triples whose predicates correspond to predicates pair specified in 'Link Policy'. For evaluating similarity of the objects, IDS adopted N-gram distance method [31]. Algorithm below describes in more detail how to evaluate the sameness of source and target subject entities. Target RDF triples, whose subject entities have been evaluated to be the same sufficiently, are stored in both 'Next Depth Linked RDF Triples' and 'Accumulated RDF Triples'.

In the same manner of 'Surface Searched RDF

```
Algorithm Sameness_Evaluation(sourceTopic, targetTopic, sourcePredicate, targetPredicate) {
    for each pair of source and target RDF triples {
        if (sourceTopic and targetTopic had not been specified in link policy as a pair){
            Target RDF triple is excluded from the evaluation; }
        else {
            if (sourcePredicate and targetPredicate had been specified in link policy as a pair) {
                if (they were the first predicate pair of the subject pair) {
                    Similarity degree of objects in source and target RDF triples becomes
                    -- the sameness degree of subject entities in the source and target RDF triples; }
                else {
                    The sameness degree of subject entities in source and target RDF triples
                    -- is reinforced by reflecting similarity degree of objects
                    -- in source and target RDF triples repeatedly; }
            }
        }
        if (sameness degree of subject entities in source and target RDF triples exceeds
            -- Sameness_Degree requested by the user) {
            Pass the target RDF triple to 'Next Depth Linked RDF Triples'; }
        else {
            Pass over the target RDF triple; }
    }
}
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Triples', RDF triples in 'Next Depth Linked RDF Triples' are furnished to 'Link Policy Consulting'. In other words, likewise RDF triples in 'Surface Searched RDF Triples', RDF triples in 'Next Depth Linked RDF Triples' take on a role of starting point of next depth searching. At this searching level of depth, IDS regards 'Next LOD' as a new source LOD and then consults a link policy of the new source LOD. 'Link Policy Consulting' looks into this new link policy and then composes 'Source RDF Triple' and 'Target Qualification'.

As presented, IDS repeats those procedures until it reaches at 'Depth_Level' requested by the user. IDS stores RDF triples, which it have gathered during the continuous in-depth searching, into 'Accumulated RDF Triples' and then hands over them to 'RDF Triple Reorganizing' in order to compose searching results.

4.2.2 SameAs link based next depth searching

From 'Surface Searched RDF Triples' in Figure 3, IDS may find an RDF triple whose predicate is <owl:sameAs>, i.e., a sameAs link triple. The RDF triple describes explicitly that its subject and object are the same. In this case, the object is an entity identified only with URI. By utilizing the URI, 'SameAs Link based Next Depth Searching' infers URI of SPARQL Endpoint. In most cases, URI of SPARQL Endpoint is a form of <http://ko.dbpedia.org/sparql> [7, 8, 10]. 'SameAs Link based Next Depth Searching' gets an RDF triple whose subject entity has been the object entity in the sameAs link RDF triple. It passes the triple to 'Accumulated RDF Triples' without hesitation. Sameness between the subject and the object has been declared by LOD publisher. It is therefore natural to regard the sameness as highly trustworthy [27]. Every RDF triple in 'Accumulated RDF Triples' will join in final searching results.

4.3 Results Reorganizing

IDS provides searching results that comply with RDF model so that it enables users to make full use of the results in the same way as using LODs. Similar researches have been presented at [32-34]. From 'Accumulated RDF Triples' in Figure 3, at first, 'RDF Triple Reorganizing' gets a subject entity S_0 whose RDF triples, surface RDF triples for clarity, have been obtained from 'Surface LOD'. For the other RDF triples which have been obtained from LODs met during in-depth searching, in-depth RDF triples for clarity, in 'Accumulated RDF Triples' whose subjects have been evaluated to be the same enough to S_0, 'RDF Triple Reorganizing' utilize predicates and objects of the in-depth RDF triples to expand the knowledge about S_0 of surface RDF triples. In other words, S_0 becomes a subject in common of predicates and objects of surface RDF triples and in-depth RDF triples. Although a user had accessed only 'Surface LOD', consequently, his/her searching request has acquired RDF triples from unexpected LODs which have described facts about the joint subject S_0 from their own viewpoints. New RDF triples of S_0 are stored in 'RDF Triple Results'.

If a user tried to utilize 'RDF Triple Results' in the same way as LODs, he/she would need ontology. IDS, therefore, prepares an ontology which has been customized for 'RDF Triple Results'. 'Ontology Customizing' accesses ontologies of LODs from which IDS had obtained RDF triples for expanding knowledge about S_0. From the ontologies, 'Ontology Customizing' perceives syntactic features of predicates. It also figures out class definitions of subjects and objects. With the information, 'Ontology Customizing' composes 'Ontology Results' which complies with RDF model as well. IDS serializes 'RDF Triple Results' and 'Ontology Results' with N-triple method [10, 12]. It is possible to regard the searching results as a new LOD. Although the new LOD is relatively small, it provides facts expanded with very various viewpoints. And what is more, it can be access by SPARQL queries.

4.4 Differences from Previous Works

IDS has three major differences from previous works [20-24]. First, IDS provides facilitations for linking identical subject entities in different LODs. Instead of generating explicitly the sameAs links and appending them to the LODs, IDS allows establishing link policy for each LOD. It lightens heavy burdens for generating and maintaining the sameAs links that exist explicitly in LODs. Second, IDS does not omit any changes in LODs which must appear in the searching result. Link policy is accessed at the time of searching and IDS takes the target entities by consulting the link policy. Unlike the previous works, this feature of link policy consulting guarantees the searching results to be up-to-date. Finally, link policy provides flexibility to prepare conditions for searching identical subject entities. The flexibility has its origins in the link policy. This research provides vocabularies set for specifying the link policies in detail. Above all the things, this research is the first proposal to prepare link policy for each LOD and to consult it at the time of searching to find identical entities.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. EXPERIMENTS AND ANALYSIS

This research has performed experiments on IDS by using LODs of DBpedia published by Korea, France, Italy, Portugal, and Spain. Numbers of subject entities in the LODs at the time of June, 2019 are presented in Table 2.

Table 2: Number of Subject Entities

LOD	Number of Subject Entities
http://ko.dbpedia.org	310,811
http://fr.dbpedia.org	1,591,318
http://it.dbpedia.org	968,794
http://es.dbpedia.org	1,120,144
http://pt.dbpedia.org	865,889

DBpedia has been regarded as the most successful LOD project [35, 36]. During the experiments, Korea DBpedia took on a role of 'Source_LOD'. Other four DBpedia LODs were specified as target LODs in link policies. Similarity degree of source and target objects was applied as 1.0, 0.9, 0.8, and 0.75. SPARQL queries for getting RDF triples about 'Iron Man' in Korean, 'Fast and Furious' in Korean, 'Spider Man' in Korean, and 'Batman' in Korean were requested to Korea DBpedia. In-depth searching progressed until it arrived at Depth_4. Korea DBpedia provided 16 entities as results. There were 64 sameAs links starting from these surface entities to entities stored in France, Italy, Portugal, and Spain DBpedia LODs. In order to proceed in-depth searching by consulting link policies of LODs, movie has been specified as a topic to confine within the link policies. Title, director and year released have been specified as predicates to be matched. With these circumstances, recall ratio and precision ratio of IDS's searching results have been analyzed.

5.1 Recall Ratio

When IDS proceeds on depth_i, recall ratio has been analyzed as Equation (1). Recall_Ratio(i) presents the ratio of the amount of subject entities which have been searched until 'Link Policy based Next Depth Searching' arrives at depth_i to the amount of subject entities which have been searched until 'SameAs Link based Next Depth Searching' arrives at depth_i.

$$\text{Recall\_Ratio}(i) = |\text{Policy\_Search}(i) - \text{SameAs\_Search}(i)| \,/\, |\text{SameAs\_Search}(i)| \quad (1)$$

- Policy_Search(i): Set of subject entities searched until 'Link Policy based Next Depth Searching' arrives at depth_i. It may include some subject entities which have been searched by sameAs links. 'Link Policy based Next Depth Searching' gets a subject entity if the entity were linked explicitly by an entity which had been searched with consulting link policy and evaluating the sameness.
- SameAs_Search(i): Set of subject entities searched until 'SameAs Link based Next Depth Searching' arrives at depth_i. Every entity has been discovered only with sameAs links which have been specified explicitly in LODs.

SameAs_Search(i) has been regarded as a base for analyzing recall ratio because explicit sameAs links have been considered as normal for specifying the sameness of entities which have different identifiers. The sameAs links, moreover, are considered as highly trustworthy because they were specified and announced by LODs' publishers. 'Policy_Search(i) – SameAs_Search(i)' in Equation (1) means that target entities searched by 'SameAs Link based Next Depth Searching' have been counted out from set of target entities searched by 'Link Policy based Next Depth Searching'. It presents target entities searched by consulting link policies entirely. Recall ratios have been analyzed from the standpoints of depth levels, number of target LODs specified in link policy, and sameness degree which a user requested. Figure 4, 5, 6 present average recall ratios with respect to combinations of the standpoints.

Sameness degree in user request has been attempted with 1.0, 0.9, 0.8, and 0.75. Experiment results from sameness degree 0.75, however, have been counted out because they provided enormous amount of subject entities which were excessively different from subject entities searched from 'Surface LOD'. In case of sameness degree 1.0, irrelevant to both the number of target LODs specified in each link policy and depth levels IDS proceeded, very little expansion of RDF triples has been realized. Sameness degree 1.0 can be accomplished only if every target RDF triple's object were exactly the same as to its corresponding source RDF triple's object. As long as sameness degree is requested to be 1.0, subject entities of target RDF triples whose objects are not exactly the same have been excluded from being searched by 'Link Policy based Next Depth Searching'. In addition, most of target RDF triples whose objects are exactly the same might have been linked with sameAs links. Equation (1) excluded these as well. Consequently, IDS does not provide any additional

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RDF triples as long user requests source and target objects to be the same exactly.

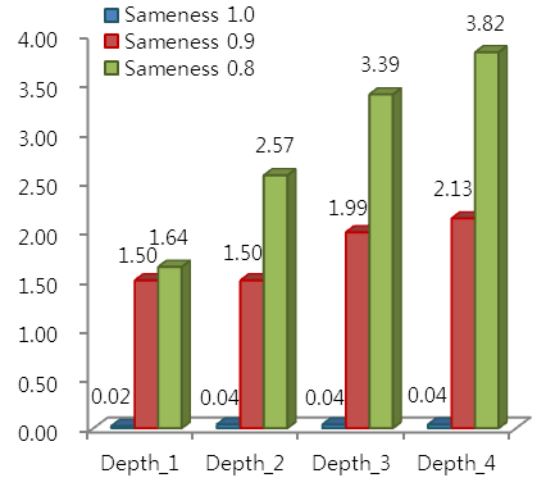

Figure 4: Recall Ratio (2 LODs in Link Policy)

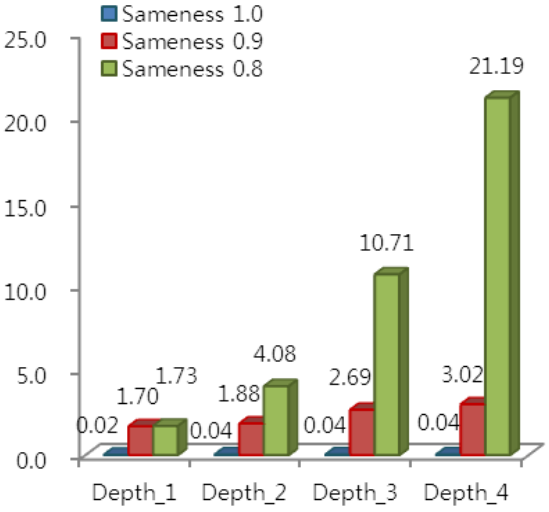

Figure 5: Recall Ratio (3 LODs in Link Policy)

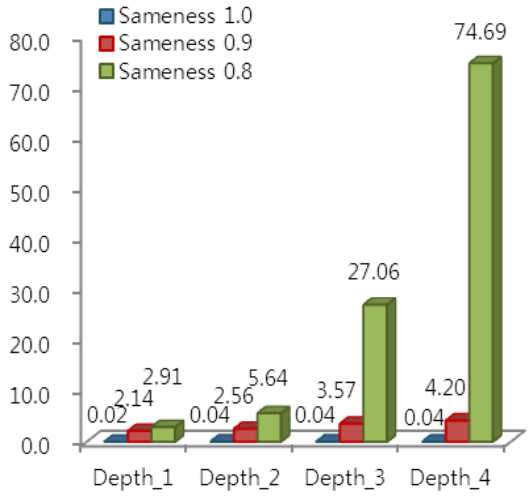

Figure 6: Recall Ratio (4 LODs in Link Policy)

For sameness degree 0.9, recall ratios increase appropriately as depth deepens and number of target LODs specified in link policy increases. Because a user alleviated the requirement of similarity of source and target RDF triples' objects, IDS could provide RDF triples which would contribute to expand knowledge obtained from source LOD's RDF triples. By observing the individual RDF triples expanded, most of them were sufficiently relevant to the surface LOD's RDF triples. Dissimilarly to 0.9, sameness degree 0.8 leads recall ratios to increase rapidly. Although similarity requirement of objects in source and target RDF triples has been alleviated slightly, increment of recall ratio responded sensitively. Searching results include considerable amount of RDF triples whose subject entities are far different from subject entities searched from surface LOD. Number of LODs in each link policy and depth level in searching request accelerated the increment of recall ratios in the case of sameness degree 0.8.

5.2 Precision Ratio

Precision ratio has been analyzed for each depth IDS proceeds as Equation (2). Precision_Ratio(i) presents the ratio of the amount of subject entities which have been searched commonly by both 'Link Policy based Next Depth Searching' and 'SameAs Link based Next Depth Searching' until they reached at depth_i to the amount of subject entities which 'SameAs Link based Next Depth Searching' has searched until it reached at depth_i.

$$\text{Precision\_Ratio}(i) = |\text{Policy\_Search}(i) \cap \text{SameAs\_Search}(i)| \,/\, |\text{SameAs\_Search}(i)| \quad (2)$$

Definitions of Policy_Search(i) and SameAs_Search(i) are identical to the definitions presented for Equation (1). Likewise analysis on recall_ratio, SameAs_Search(i) has been regarded as the base for precision ratio. In LOD cloud, DBpedia LODs are recognized to be highly trustworthy. The sameAs links explicitly specified within them, therefore, are regarded trustworthy as well. Entity set of SameAs_Search(i) which have been obtained by following the sameAs links can be qualified as a base for analyzing the precision ratio.

For the analysis, standpoints such as depth levels, number of target LODs in link policy for each LOD, and sameness degree have been considered. Figure 7, 8, 9 present average precision ratios with respect to combinations of the standpoints. At the standpoint of number of target LODs in a link policy, unlikely to recall ratio, precision ratios of 2, 3, and 4 target LODs do not show huge difference.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This is due to special circumstances of DBpedia LODs. In DBpedia LODs, unlikely to other LODs in LOD cloud, sameAs links have been prepared with dense and symmetric. The number of target LODs in link policy, therefore, does not have influences on the increment of precision ratios.

As for the depth levels, precision ratios increased according as in-depth searching proceeds deeply. Precision ratios of sameness degree 0.9 increased relatively high when IDS proceeded from depth_2 to depth_3. The increment became slow as IDS proceeded from depth_3 to depth_4. Concerning the sameness degrees, as a user relaxed the sameness requirement, precision ratios increased accordingly. Especially for the sameness degree of 0.8 and 0.75, IDS searched abundant entities, and precision ratios eventually approached to almost 1.0. However, the results included considerable amount of noisy entities. Therefore, sameness degree which has been requested below 0.8 is regarded as inappropriate.

In case of sameness degree 0.9, although the amount of entities searched was considerably small, comparing those of sameness degree of 0.8 and 0.75, it has led precision ration close to 0.9. As we have investigated during recall ratio analysis, sameness degree 1.0 does not contribute to expand searching results. We anticipated, therefore, that a lot of in-depth searching would be blocked and precision ratio would decrease accordingly. Contrary to the expectations, at depth_3 and depth_4, precision ratios of sameness degree 1.0 became close to those of sameness degree 0.9. These situations inform that link policy based in-depth searching had compensated a lot of the entities which would have been searched in a way of following the sameAs links.

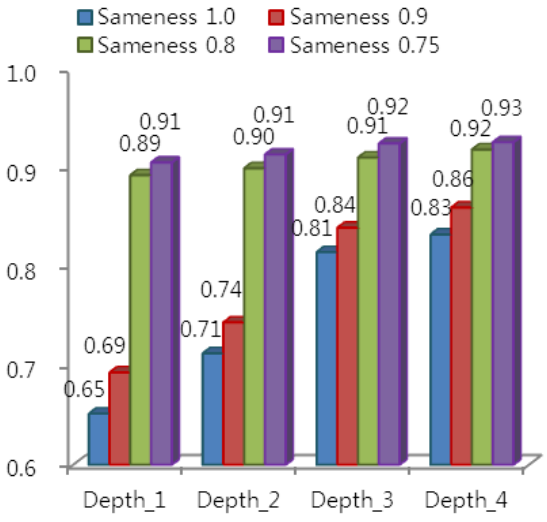

Figure 7: Precision Ratio (2 LODs in Link Policy)

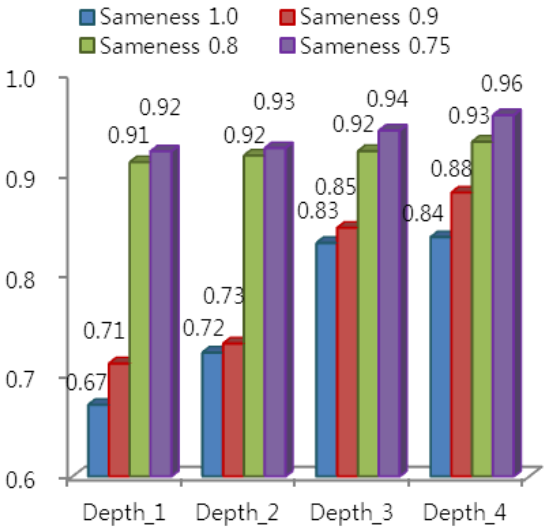

Figure 8: Precision Ratio (3 LODs in Link Policy)

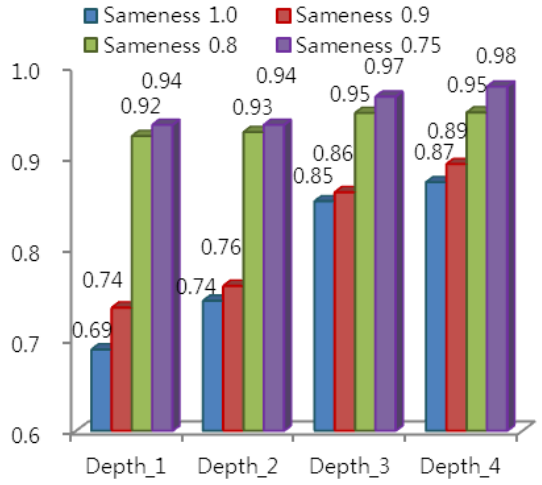

Figure 9: Precision Ratio (4 LODs in Link Policy)

5.3 Analysis Summary and Recommendations

For the number of target LODs to specify in link policy, recall ratio increased almost in proportion if a user requested sameness degree as 0.9. To achieve appropriate recall ratio, LOD publishers are recommended to specify three or four target LODs in their link policies. For users in LOD cloud, sameness degree is recommended to be requested between 0.8 and 0.9. If a user requested it as close to 1.0, recall ratio would be almost zero and thus knowledge expansion would be impossible. If sameness degree were requested as below 0.8, although it would provide almost perfect precision ratio, searching results would expand excessively at depth_3 or depth_4 with large number of noisy entities. For achieving meaningful precision ratios, users are recommended to request depth level as 3 or 4 as long as sameness degree is 0.9. If sameness

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

degree were requested close to 0.8, it would be desirable to request the depth level as to be 3.

6. CONCLUSIONS AND FUTHER WORKS

This research suggested that LODs' publishers would prepare their own link policies and publish them together with their LODs. The suggestion will relieve heavy burdens of specifying sameAs links between source and target entities to achieve knowledge expansion, an essential goal of LOD cloud. Instead of following the sameAs links, by consulting the link policies to find identical target entities at the time of searching, searching results will not omit any modifications in LOD cloud. To establish the link policy, this research devised a set of vocabularies which would enable publishers to specify their link policies in detail. Likewise LODs, link policies are composed of RDF triples and thus can be accessed by SPARQL queries.

In order to convince the appropriateness of consulting link policies during searching in LOD cloud, this research implemented In-Depth Searching (IDS) system. IDS at first finds entities from surface LOD. Then it proceeds in-depth searching to target LODs and obtains entities by consulting link policies. Link policies provide information about target LOD to access next and set of source and target predicate pairs which have been specified under a topic for confining candidate entities. For evaluating sameness between source and target entities, the target entity must be confined to have a topic specified in source LOD's link policy. The target entity, in addition, must have predicates specified in source LOD's link policy as to be pairs. If these preconditions were satisfied, IDS would regard the similarity degree of source and target objects as the sameness degree of source and target subjects. If there were two or more predicate pairs, IDS would reinforce the sameness degree. A subject entity searched from surface LOD became a joint subject entity of RDF triples obtained from source LOD and other LODs which IDS passed through during in-depth searching. The RDF triples become searching results. To enable users to access the searching results in the same way to access LODs, IDS customized an ontology specialized for the searching results.

Experiments on IDS were performed on DBpedia LODs known as the most successful LOD project in LOD cloud. From the experiments, establishing and consulting link policies have been verified to be effective for expanding knowledge with sufficient trustworthiness. In the experiments, IDS processed queries about four popular movies. Korea, France, Italy, Portugal, and Spain DBpedia LODs were participated to the experiments. Analysis was performed on recall ratio and precision ratio. The amount of entities searched with only sameAs links took a role of base for the ratios because the sameAs links of DBpedia were recognized to be highly trustworthy. The analysis have confirmed that a meaningful recall ratio can be achieved as long as 3 or 4 target LODs are specified in a link policy, a user requests sameness degree as to be between 0.8 and 0.9, and in-depth searching proceeds until depth_3 or depth_4. For appropriate precision ratio, users are recommended to request sameness degree as to be around 0.9. For depth level, they are recommended to request IDS to proceed in-depth search until it arrive at depth_3 or depth_4.

In further works, in order to evaluate the sameness more elaborately, syntactic features of predicates in source and target RDF triples need to be considered. Especially, since OWL2 provides plentiful inference features, it is anticipated to obtain many opportunities to improve trustworthiness of sameness evaluation results. With these works, vocabulary set in Table 1 is expected to be complemented and thus to improve its expressiveness. [37, 38] will provide valuable methods to devise new vocabularies. Reputation of LOD is worthy of considerations. It is expected to play a meaningful role in reinforcing sameness degree between source and target entities. For evaluating the reputations, incoming and outgoing links of LOD need to be measured. Another important topic of further works is to find out the feature of distance and agreement of sameAs links. By following the sameAs links acquired during IDS proceeds, a graph, in which sameAs links and subject entities compose edges and nodes, will appear. As a sequence of sameAs links becomes longer, sameness degree of a subject entity which is at distance becomes faint. In the further works, sameness evaluation will reflect this feature. On the other hand, if cycles appeared in the sameAs link graph, since there were agreements on the sameness, sameness degree would be reinforced. Further works will concentrate on improving trustworthiness of sameness evaluation.

7. ACKNOWLEDGEMENT

This research was supported by Seokyeong University in 2018.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] W3C, "What is Linked Data, 2018," 2018; https://www.w3.org/standards/semanticweb/data

[2] D. Stuart, Practical Ontologies for Information Professionals, Facet Publishing, 2016.

[3] E. F. Kendall and D. L. McGuiness, Ontology Engineering (Synthesis Lectures on the Semantic Web: Theory and Technology), Morgan & Claypool Publishers, 2019.

[4] J. Abbas, Structures for Organizing Knowledges: Exploring Taxonomies, Ontologies, and Other Schema, Neal-Schuman Publishers, 2010.

[5] R. Arp, B. Smith, and A. D. Spear, Building Ontologies with Basic Formal Ontology, MIT Press, 2015.

[6] A. Abele and J. McCrae, "The Linked Open Data cloud diagram, 2018," 2019; http://lod-cloud.net/

[7] S. Sakr, M. Wylot, and et al., Linked Data: Storing, Querying, and Reasoning, Springer, 2018.

[8] D. Wood, M. Zaidman, and et al., Linked Data: Structured Data on the Web, Manning Publications, 2014.

[9] M. C. Daconta, The Great Cloud Migration: Your Roadmap to Cloud Computing, Big Data and Linked Data, Outskirts Press, 2013

[10] S. Auer, V. Bryl, and S. Tramp, Linked Open Data – Creating Knowledge Out of Interlinked Data: Results of the LOD2 Project, Springer Open, 2014.

[11] A. Harth, K. Hose. And R. Schenkel, Linked Data Management (Emerging Directions in Database Systems and Applications), Chapman and Hall, 2014.

[12] N. Konstantinou, Materializing the Web of Linked Data, Springer, 2015.

[13] L. F. Sikos, Mastering Structured Data on the Semantic Web: From HTML5 Microdata to Linked Open Data, Apress, 2015.

[14] J. E. L. Gayo, E. Prudhommeaux, and et al., Validating RDF Data, Morgan & Claypool Publishers, 2018.

[15] C. Bizer, "Is the Semantic Web what we expected, 2017," 2017; https://www.slideshare.net/bizer/is-the-semantic-web-what-we-expected-adoption-patterns-and-contentdriven-challenges-iswc-2016-keynote

[16] A. Harth, Linked Data Management, CRC Press, 2014.

[17] Z. Jeff and P. V. Guido, Exploiting Linked Data and Knowledge Graphs in Large Organizations, Springer, 2017.

[18] B. Valentina, Combining Truth Discovery and RDF Knowledge Bases to Their Mutual Advantage, in *Proceedings of the 17th International Semantic Web Conference*, Monterey, Canada, 2018, pp. 652-668.

[19] D. Allemang and J. Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kaufman, 2011

[20] A. Ngonga and S. Auer, "LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data," in *Proceedings of the 22nd IJCAI*, Catalonia, Spain, 2011, pp. 2312-2317.

[21] J. Volz, "Silk – A Link Discovery Framework for the Web of Data," in *Proceedings of the 2nd Workshop on Linked Data on the Web*, Madrid, Spain, 2009, pp. 238-247.

[22] J. Park and Y. Sohn, "A Syntax Added Link Evaluation Technique for Improving Trustworthiness of LOD's Linkages," *Journal of KIISE: Databases*, vol. 41, no. 1, 2014, pp. 45-61.

[23] J. Park and Y. Sohn, "Trustworthiness Improving Link Evaluation Technique for LOD Linkages giving Considerations to the Syntactic Properties of RDFS, OWL, and OWL2, " *Journal of KIISE: Databases*, vol. 41, no. 4, 2014, pp. 226-241.

[24] Y. Sohn, "Reliability Improving SameAs link Evaluation Technique for Linked Open Data Publication," *INFORMATION*, vol. 19, no. 9, (2016), 2016, pp. 4271-4279.

[25] L. Boneva, Semantics and Validation of Shapes Schemas for RDF, in *Proceedings of the 16th International Semantic Web Conference*, Vienna, Austria, 2017, pp. 104-120.

[26] V. Svitlana, Measuring Semantic Coherence of a Conversation, in *Proceedings of the 17th International Semantic Web Conference*, Monterey, Canada, 2018, pp. 634-651.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[27] S. Brin, et al., "The PageRank Citation Ranking-Bringing Order to the Web," http://ilpubs.stanford.edu/422/1/1999-66.pdf, 1998.

[28] A. Yamaguchi and H. Toh, Implementing LOD Surfer as s Search System for the Annotation of Multiple Protein Sequence Alignment, *Proceedings of JIST: Joint International Semantic Technology Conference*, Hyogo, Japan, 2018, pp. 418-426.

[29] B. DuCharme, *Learning SPARQL*, O'REILLY, USA, 2013.

[30] W. Meng, Towards Empty Answers in SPARQL: Approximating Querying with RDF Embedding, *Proceedings of the 17th International Semantic Web Conference*, Monterey, Canada, 2018, pp. 513-529.

[31] B. Peter, "Class-based n-gram models of natural language," *Computational linguistics*, Vol. 18, No. 4, 1992, pp. 467-479.

[32] P. Vayianos, Ontology Driven Extraction of Research Processes, *Proceedings of the 17th International Semantic Web Conference*, Monterey, Canada, 2018, pp. 162-178.

[33] E. Andaroodi and F. Andres, Ontology-based Semantic Representation of Silk Road's Caravanserais: Conceptualization of Multifaceted Links, *Proceedings of JIST: Joint International Semantic Technology Conference*, Hyogo, Japan, 2018, pp. 89-103.

[34] A. Soylu and E. Kharlamov, Making Complex Ontologies End User Accessible via Ontology Projection, *Proceedings of JIST: Joint International Semantic Technology Conference*, Hyogo, Japan, 2018, pp 295-303.

[35] T. Kawakami, T. Morita and T. Yamaguchi, Building Wikipedia Ontology with More Semi-structured Information Resources, *Proceedings of JIST: Joint International Semantic Technology Conference*, Gold Coast, Australia, 2017, pp. 3-18.

[36] P. Lertvittayakumjorn, N. Kertkeidkachorn, and R. Ichise, Resolving Violation in DBpedia, *Proceedings of JIST: Joint International Semantic Technology Conference*, Gold Coast, Australia, 2017, pp. 121-137.

[37] S. Joo, H. Takeda, and et al., Building the Core Vocabulary of Crop Names to Integrate the Vocabularies by Multiple Government Agencies, *Proceedings of JIST: Joint International Semantic Technology Conference*, Hyogo, Japan, 2018, pp. 320-335.

[38] D. Ratcliffe and K. Taylor, Refinement-based OWL Class Induction with Convex Measures, *Proceedings of JIST: Joint International Semantic Technology Conference*, Gold Coast, Australia, 2017, pp. 49-65.