On Improving SPARQL Information Retrieval System for Semantic Internet of Things Applications

Abdulrahman Jalal
King Abdulaziz University, Jeddah, Saudi Arabia

Sofiane Ouni
University of Manouba, Tunisia

Karim Kamoun
Higher School of Digital Economy of Manouba, Tunis

Bassam A. Zafar
Faculty of Computing and Information Technology King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT

Due to the huge volume of information and knowledge derived from IoT devices, it will be hard to explore all knowledge coming from these devices. The semantic ontology-based descriptions with the semantic web are one of the most interesting ways to extract the required knowledge. However, the information retrieval typically made by the SPARQL querying language stills hard to write correct queries from what the users need as knowledge to extract. It needs a deep knowledge of the semantic information system structure.

In this paper, we have developed a new correction and relaxation approach based on structural semantic similarity measure to overcome the semantic errors in SPARQL queries. This approach is applied to semantic information systems using OWL and RDF ontologies which are related to IoT applications. To achieve the efficiency of our proposal, we have developed a SPARQL querying tools. According to the queries made on IoT applications, our approach performs best results regarding the precision of the answer to these queries.

General Terms

Information Retrieval, semantic similarity, Internet of Things, SPARQL

Keywords

Ontology, Garden Smart Park, Sensor

1. INTRODUCTION

Internet of things (IoT) is the ability for things that contain embedded technologies to sense, communicate, interact, and collaborate with other things, thus creating a network of physical objects. In recent years this concept has gained enormous momentum and is now one of the most talked about things in the world of today technology. At this rapid rate of growth, it is projected that there will be approximately 26 billion connected devices by 2020 [1].

The IoT community requires coordinated efforts to define more vocabularies and description frameworks to represent resources, data, and services in the IoT domain [2]. Also, there was a need felt for synergetic efforts from other fields such as service computing, data mining, ontology queering, information retrieval and knowledge extraction and exploration to enhance the processing and utilization of semantic data in the IoT domain [3].

Due to the increase and the huge number of information and knowledge derived from IoT devices, it will be difficult to easily find what we need as information from IoT devices and sensors [4]. Devices and sensors can read data, but then they typically seeking keywords to find information, machines have

difficulty extracting any meaning from these data themselves [5]. The data access which provides connections to Big Data storage for the collected data from devices will be using semantic query requests [6]. The typical querying systems having proposed for ontology is SPARQL that requests information from ontologies written in RDF and OWL. It is inspired by SQL querying for relational database [7]. However, writing a correct query with SPARQL is not an easy task. It needs to know the structure of the inquired ontology. It is difficult to be informed about all concepts in the ontology. That's way, we need to be helped by some tools and procedures to correct and improve the queries. These tools should be based on the similarity measures for getting the right and nearest semantic meaning according to the query.

In this paper, we propose a new way for improving the correctness of information retrieval based on SPARQL for IoT with ontology description. It is based on content and structural similarity measure to get the best correspondence of the query to inquired ontology. This approach should extend the related works on query relaxation and rewrite. In spite of the query improving results made by these approaches, they still limited in many cases especially when the original query is with some semantic errors.

The following paper is organized into 6 sections. In the second section, we will have a background about the semantic internet of things, semantic information retrieval and querying with SPARQL. The third section will mention some of the related works for SPARQL extensions. After that, we will offer our contribution to overcoming the limits of the related works. Then we will have a case study to evaluate our contribution and present the evaluation results. Finally, we will have the conclusion of this paper.

2. SEMANTIC NOTIONS

2.1 Semantic Internet of Things

The international community World Wide Web Consortium (W3C) has produced the Semantic Sensor Network ontology (SSN) [8]. This ontology is mad to describe the sensors and their observations how it works, its state the involved procedures, the interest studied features, and what is the properties that been observed [9], as well for actuators (fig. 1). Regardless of SSN scopes and degrees of axiomatization, it is able to support a large scope of use cases and applications, for example the Web of Things, large-scale scientific monitoring, observation-driven ontology engineering, social sensing, citizen science, and the infrastructures for household and industry, and satellite imagery. And it helps in retrieving the desired information from any query system.
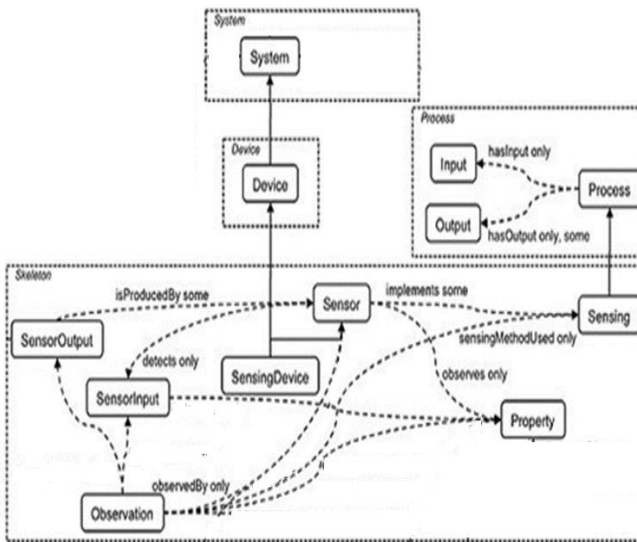
Figure 1: SSN sematic description of IoT devices

2.2 Querying with SPARQL

SPARQL is defined as a Resource Description Framework (RDF) query language, which is a semantic query language dealing with retrieving, store and manipulating data in the databases in a Resource Description Framework format, able to retrieve and manipulate data stored in that database [10].

A SPARQL query comprises [11], in the order (fig. 2):

- Prefix declarations, for abbreviating URIs

- Dataset definition, stating what RDF graph(s) are being queried

- A result clause, identifying what information to return from the query

- The query pattern, specifying what to query for in the underlying dataset

- Query modifiers, slicing, ordering, and otherwise rearranging query results

Figure 2: SPARQL Query structure

2.3 Semantic Similarity Measures

The semantic similarity measure is provide the similarity between two concepts according to specific context that gives the meaning of the concepts in use [12].

WordNet [13] is one of famous databases of word synonyms using semantic similarity measures based on a great number of ontologies. It is defined as a lexical reference system based online, first developed at Princeton University. WordNet put effort to constructing the lexical knowledge of a native speaker for the English language. Also WordNet can be counted as an ontology that supported natural language terms. It comprises of over than 100,000 terms which constructed in taxonomical hierarchies [14]. The WordNet includes Nouns, verbs, adjectives and adverbs which grouped into synonym sets (synsets) [15]. Also the synsets are organized into something called senses that's means corresponded to different meanings of the same concept or term. The synsets can be related to other synsets that are higher or lower in the taxonomy hierarchy. However, related to the similarity measures, WordNet still consider as a lexical tool. So, in specific context of ontology description, it cannot provide a precise and accurate measure that way we need specific similarity measures that are related to the context focused on. Many related works make a mixing of another similarity with WordNet to get more complete measures [16] [17] [18].

In general, there are two main kinds of semantic measure to use for measuring the concepts semantically which are Corpus-based and Knowledge-based measures.

Corpus-based measures used to compare words of a language, concepts or instances from text analysis i.e. structured semantic proxies. These measures are typically used to compare words of a language.

Knowledge-based measures are designed for comparing entities defined in ontologies, i.e. structured semantic proxies. Comparing words of a language, it can be done through using Knowledge-based measures for establishing bridges between texts and ontologies. The Knowledge-based measures [19] has three measures under it which they are Information Theoretical, Feature-based and Structural Approach which we will use here. Structural measures rely on graph-traversal approaches [20]. They focus on the analysis of the interconnection between concepts to estimate their similarity. In this context, some authors [21] have proposed an approach based on graph theory [22]. So, the similarity will be defined relative to concepts links in the ontology graph.

3. SPARQL EXTENSIONS RELATED WORKS

In this section, we describe the most relevant paper that extends the SPARQL querying for improving the obtained results.

The paper [23] deal with the semantic web techniques for retrieving the most relevant information related to the user recommendations by using an extending SPARQL querying called RecSPARQL. The SPARQL extension is based on the recommender system so that suggests recommenders and SPARQL can be combined by using SPARQL for processing the recommender's input data, respectively output data.

The Recommender systems typically produce a list of recommendations through collaborative filtering. It approaches to build a model from a user's past behavior as well as similar decisions made by other users. This model is then used to predict items that the user may have an interest in. These items will be a part of the main data that build the SPARQL queries. Thus, RecSPARQL uses these data to extend, and make a generic and flexible query.

Figure 3: Query relaxation

Another extension for SPARQL is QaRS: A User-Friendly Graphical Tool for Semantic Query Design and Relaxation. This paper [24] presents a Query-and-Relax System (QaRS) designed cooperative techniques help users in finding alternative answers when their queries do not return the expected number of answers. The proposed includes 3 ways for relaxations related to the difficulties of the relaxation problem. It can be made automatically where the system automatically relaxes the query based on similarity measures. The relaxation can be by selecting more general information, classes and properties in the query to be replaced (fig. 3). This automatic relaxation is evaluated, if it does not give the required answers by an evaluation module, the user will use a manual and interactive relaxation.

In these related works about semantic information retrieval and querying, there were many limitations according to query accuracy and result retrieval efficiency. One of the main problems of the related works is the shortage of the result for a specific query when the constraints of the query prevent to get all the possible outcome form the ontology. The solution to this problem is to remove as low as possible of the constraints in the query to get all the desired result, and this process is called relaxation. The relaxation process depends on relaxing the query by removing the query constraints, so that will help to get more result but with low present of precision. Our goal in this paper is to propose an extended querying system based on similarity measures to help in efficient and precise information retrieval.

From this presentation of related work, the SPARQL semantic-based information retrieval system is one of the most relevant issues. But it is inefficient to retrieve information from query system with high precision. The next section will discuss some solution to improve the SPARQL for the need of information retrieval.

4. OUR CNTRIBUTION FOR MPROVING QUERY SYSTEM
4.1 IoT Querying System Architecture

The semantic System architecture as presented in (fig. 4) is constructed on four main Modules: IoT platform, Ontology information system, query system and back-end applications.

The first module is the IoT platform that collects the data from IoT devices or sensors and forwarded through a dedicated wireless network to the gateway and to the cloud system where we can find databases as a semantic description or ontology. The importance of the IoT Platform is relying on some features which will help the system to be a masterpiece. The features that will be provides when employing an it platform to the system are like; first is management and integration of the IoT devices and handle configuration, firmware updates and provide device-level error reporting and error handling, second is keeping the information secure from any risk, and encrypted it to avoid potential eavesdropping, third is data collection protocol and this is an important feature since it specifies the types of protocols used for the data communication and transfer between the components of an IoT software platform. The IoT platform varying on what feature they provide, some of the most used platforms are like AWS IoT platform, Bosch IoT Suite - MDM IoT Platform, 2lemetry - IoT Analytics Platform, EVRYTHNG - IoT Smart Products Platform [25].

The second module is the ontology information system where all the triples are stored for the query usage by the end user when there is the need for querying an information, the ontology is controlled by the ontology creator, and ontology manager who is responsible for updating expanding or deleting from the ontology.

The third module is the query system which is responsible for responding to the end user query and translate it to the ontology to retrieve the desired information from the ontology to the end user, the query system is composed of three parts (result query evaluation, query correction and relaxation and query list). The result query evaluation is to evaluate the returning result of the query from the user query. The query correction and relaxation process take place if the returned query result is none or not sufficient. Then, there will be a correction and relaxation of the querying parameter to retrieve wide and more range of result for the query. All approved queries are added to the query list in order to provide to the end user some predefined valid queries.

The fourth module is the back-end applications which is run by the end user to execute a query in order to retrieve desired information. It can be also some applications with specific interfaces created to display the data carried out from the ontology by using the new efficient querying system.
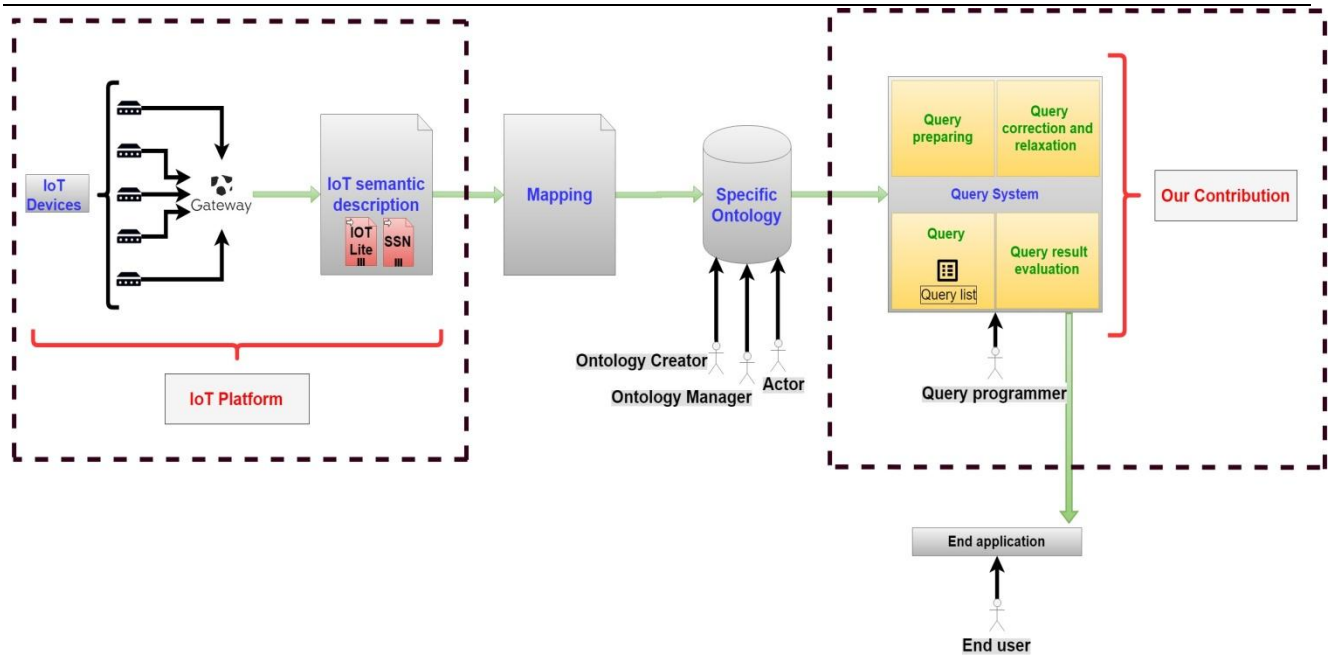
Figure 4: SPARQL querying system for IoT Query correction procedure

The main contribution of the SPARQL querying system that we have designed is the Querying Correction and Relaxation procedures. It makes the correction of the wrong query which gives no result. The error on the query can be related to the syntax and the semantic description. The syntax is corrected by the used tools of running the queries. But the semantic is not treated due to the complexity and diversity of such thing. So, in this paper, we propose a way to correct the main semantic errors that can be made. It is mainly related to using of incorrect terms from the ontology. This kind of error can often occur because of the great number of concepts and terms that can be used in the ontology. So, the user can be confused and write the query with terms or concepts that are not provided by the ontology. In this case, no result will be provided and the lexical correction system will not give errors. That's why the user will be confused and do not know what the problem is. In this case, our proposed correction procedure will guide and assist the user to correct semantically his query.

The correction will be by getting the right terms to replace or substitute the wrong ones in order to give the desired result. While in this case of incorrect query, the relaxation procedures used in related work cannot give the right answer to this kind of error.

In the case of incorrect query, we need to apply the correction procedure that makes the substitution of the incorrectly used terms within the similar term in the ontology to the query context. This latter gives the semantic description of the query. Thus, we need to understand the meaning of the query by a semantic description of the related used terms in this query. The correction will be based on this description and by retrieving the nearest term in the ontology that can be well mapped to this query. For that, we will do a similarity measure based on the structural description and by the way the graph structure of the query.

4.2 Query Graph Modeling
The query graph gives the semantic meaning of the query terms through the relation between concepts. It helps to find the incomplete or incorrect used terms of the query to be

substituted from the ontology through using semantic similarity measure [26].

Ontology query is a graph structure for semantic queries with nodes and edges. Here is what nodes, and edges mean, and how it relates to the ontology query graph:

- Nodes: They represent entities or concepts.

- Edges: also known as relationships or graphs, are the lines that connect between the nodes, these edges represent the relationship between the nodes. We can say edges are the main key that interprets the semantic through the link of data.

4.3 Similarity Measure for the Procedure
Fig. 5 shows a simple for finding the word to replace the wrong one ($\overline{CY}$) in the query. It starts by modelling the query into query graph. Then, we get the best-matched word to $\overline{CY}$ from the ontology. This found word will substitute $\overline{CY}$.

By the use of similarity measure, we want to have the best matching of the terms of the query to the concepts inside the inquired ontology. This means that for a query where we are looking for the substitution of the wrong word $\overline{CY}$, we are not limited to find only the similar word in the ontology to $\overline{CY}$. But we are also focusing on the other concepts of the query related $\overline{CY}$. Thus, the new similarity measure (called: Sim_{WO}) need to consider also the similarity of all query terms, so that we can get more the semantic into the similarity measure. The impact of other terms will be related to their information content cohesion of the terms in the ontology. This cohesion is made by the use of the Wu & Palmer similarity measure.
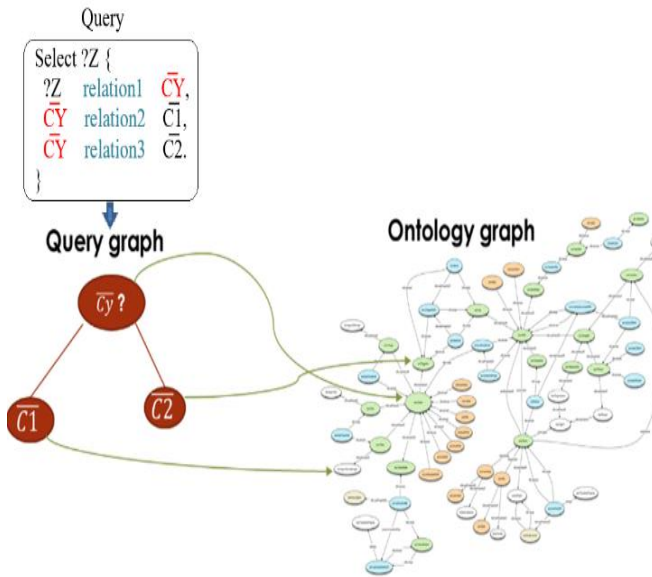
Figure 5: Querying matching to the inquired ontology

We defined a new similarity measure; called Sim_{WQ} (as the similarity for a word in the query) that makes the substitution of $\overline{CY}$ by the term that is similar to it from the ontology and also it has a semantic relation with *CX* (other concepts in the query) as in We are looking for the nearest (maximum similarity) concept in the ontology called C_i, so that:

$$Sim_{WQ}(\overline{Cy}, C_i) = \left(Sim_{WordNet}(\overline{Cy}, C_i) + \sum_{\bar{C}_k \in QueryGraph} Max_{Cj \in Ontology}\left(Sim_{WordNet}(\bar{C}_k, C_j) * Sim^*_{WuP}(\bar{C}_k, C_j) \right) \right) / NTerm \quad (1)$$

With $NTerm =$

$$1 + \sum_{\bar{C}_k \in QueryGraph, \, C_j^*} Sim^*_{WuP}(\bar{C}_k, C_j^*)$$

Where *NTerm* is a normalization term used to get the similarity value in the interval [0,1], C_j^* is the select C_j from the maximum (*Max*), $\overline{CY}$ is the word term in the query, C_i is the term in the enquired ontology. $Sim_{Wordnet}$ is the similarity measure made by WordNet and Sim_{WuP} is the similarity measure that made by Wu and Palmer. The latter is used as a weight to give the importance of the similarity measure related to the ontology structure. That is if the concept looking for is near to the other concepts in the query and the ontology according to the structure, this similarity will be higher given the best chose.

The Wu & Palmer similarity measure (called: WuP) [27] [28] [29] [30] is based on the information contents. It is related to the structure of the ontology or the hierarchy of concepts. This structure is called taxonomy that defines the class hierarchy as the super-classes or subsumption or also parent/child class relations. The WP measures the distance of the concept to their nearest common parent. Thus, it [17] calculates relatedness by considering the depths of the two concepts or "synsets" in the taxonomy, along with the depth of the LCS (Least Common Subsumer). The basic formula that measures the WuP similarity between the two concepts C_1 and C_2 is:

$$Sim_{WuP}(C_1, C_2) = \frac{2 * depth(LCS)}{depth(C_1) + depth(C_2)}$$

(2)

This measure is generalized to more kind of relations between concepts that can be not only subsumption relation but also to the object properties relation between concepts [27]. Thus, the formula of the generalized Wu & Palmer similarity measure is:

$$Sim^*_{Wup}(C_1, C_2) = Sim_{Wup}(C_1, C_2) + \alpha * \frac{\sum_{prop \in E_{prop}(C_1, C_2)} Min(depth(C_1), depth(C_2))}{\left| E_{prop}(C_1, C_2) \right|}$$

(3)

where *Eprop* is the set of existing properties between concepts C_i and C_j directly or through their ancestors and α is a weighting parameter which reflects the importance level of information content regarding the property relationship compared to the inheritance information content.

5. CASE STUDY AND EVALUATION
5.1 Case Study of Smart Garden Park
We need to have an implementation and evaluation based on case study for our approach of SPARQL extensions to improving retrieving information semantically.

Our case study is a part of smart city applications [31]. It is a smart Garden Park which has different types of IoT sensors that help in gathering information from the garden park and helping the users to query about anything that related to the garden park, for example; the users can query about the weather temperature, the available restaurants, the nearest available toilet, if the running track or playground is crowded or not.

In our case study of smart Garden Park (fig. 6), there are three main classes which are the objects, the physical objects related to the IoT objects and the sensors. Each one of them has subclasses and relations with object properties, the subclass of the objects classes is person, the subclasses of the physical object class for instances garden_seats, gps_locator, play_grounds, restaurants, running_track, tanning area, fishing area, while the subclasses of sensors are fishing_area_crowdeng_sensor, restaurant_available_sensor, garden_availability_seat_sensor. Etc. For instance, the sensor garden_availability_seat_sensor is attached to the garden seat to sense either the seat is available or not.

Figure 6: Class ontology design

5.2 Simple of Related Query and Correction

In the following query, a user wants to retrieve information about all the garden seats which is available and in shadow. This information is sensed from IoT sensor **o:garden_seats**, with the condition of **"true"** for sensor garden seat availability, and the **"true"** condition for the chair to be in a shadow. Thus, the following query will display all available **seats** with the belonging **zones** and also with **sensor name**.

- Before correction

```
SELECT ?available_garden_seats  ?sensor_name  ?zone
WHERE {
  ?available_garden_seats   a                    o:garden_seats.
  ?available_garden_seats   o:sensor_available_data        "true".
  ?available_garden_seats   o:sensor_shadow_data           "true".
  ?available_garden_seats   o:has_sensor_names     ?sensor_name.
  ?available_garden_seats   o:component_of_zone           ?zone.
}
```

The previews query has an error on object property "has_sensor_names" which is stored in the ontology as has_sensor. This error will not provide a result for the query and no error is detected by the classical SPARQL querying system. However, in our correction procedure, we are able to detect and automatically correct such semantic errors. Our procedure will make a correction based on substituting the wrong terms with correct one by check the ontology semantically and compute the highest degree of similarity between the wrong term and the term in the ontology. Then substitute the wrong term by the highest similar term in the ontology. Now, this new query will provide results from the enquired ontology.

- After correction

```
SELECT ?available_garden_seats ?sensor_name ?zone
WHERE {
  ?available_garden_seats  a    o:garden_seats.
  ?available_garden_seats  o:sensor_available_data        "true".
  ?available_garden_seats  o:sensor_shadow_data          "true".
  ?available_garden_seats  o:has_sensor               ?sensor_name.
  ?available_garden_seats  o:component_of_zone         ?zone.}
```

5.3 Evolution Results

For the evaluation of the extended SPARQL querying system, we have developed an application tool for the query correction and relaxation. It is developed in JAVA language using Apache Jena API as a library for getting access to OWL and RDF ontology files, and SPARQL querying. We have used also the Java WordNet Library (JWNL) for the WordNet similarity measure based on its database.

Based on the developed application, we have created a set of queries related to the ontology of smart garden park case study, and then applied the modification related to our and other approaches in order to evaluate the queries results. For that, we have made a set of reference queries. From these reference queries, we have generated different mistakes in terms of the queries in the way that we obtain some terms different from what we can find in the inquired ontology. Then, we apply correction procedures and we look at the results if they match to the results of the reference query. So, if the correction procedure gives similar results to the reference, we can say that this approach is an interesting one. Thus, we can compute the precision based on the obtained results for our correction procedure and for the other related works based on the relaxation procedure.

The evaluation metric that we have used is mainly the precision of the obtained results according to the reference. So, the Precision (P) is the fraction of the retrieved information those are relevant according to the reference query:

$$Precision = \frac{relevant\ data\ retrieved}{retrieved\ data}$$

(4)

In (Fig. 7), we show the difference of the procession between our approach of correction and relaxation, and the other approaches that use the only the relaxation [23] [24] [32]. The value of the difference is in the interval [-1, 1].

The positive value of the difference will reflect that our approach has higher precision results. So, we note that the precision of our approach performs better results than the others related works relaxation approaches because of the obtained positive difference. For instances, in query number 1 our approach is more precise in regard to the retrieved information for the query by 0.27% (as the difference is about 0.27 up to 1), in the Query number 2 our approach give more precision than the other related works relaxation by 0.60% and so for the rest of the queries. However, in Query 5, the difference is 0. This query is a short one with a reduced number of the lines and by the way of reduced information restrictions. This means that our approach gives better results for the complex query with relevant information restrictions for clear semantic query description.
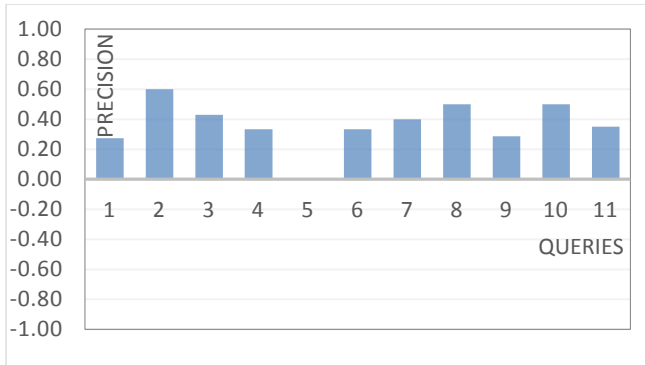
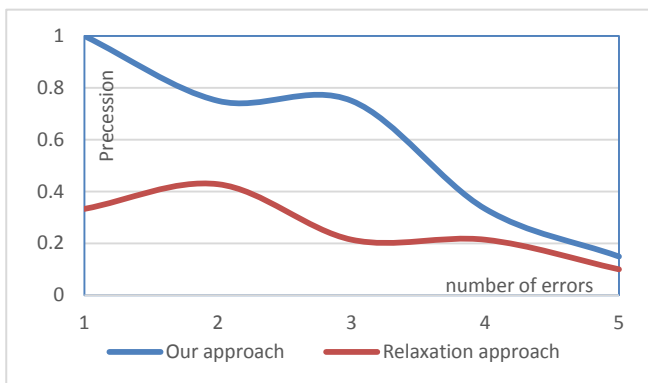**Figure 7: Relative Precision of relaxation approaches to our approach**

Figure 8: Precision variation according to the number of errors

In Figure 8, we explore the impact of the number of errors in the query to the precision of our correction approach and the other approaches. For the queries with only one error, our approach has the precision near the ideal value 1 (as %100), while the precision of the approach is about 0.33%. In the case of two errors on the query, our approach gives a precision of 0.75%, while the precision of the others is 0.42%. In the case of four errors, the difference will be less important and less efficient. Thus, with the increase of the number of the errors on the query the precision for our and others approach are decreasing, but our approach still has the competitive advantage of the higher present of precision. The decrease of precision decries the fact that if we have many errors in the query the similarity measure to map the best matching of the wrong terms to the ontology is hard to make.

6. CONCLUSION

One of the important issues related to the use of the semantic and ontology-based information system is how to efficiently explore and extract knowledge. Many scientific researchers have been interested in how to improve the SPARQL querying. They have mainly focused on the relaxation approaches which to expend the given results to more general answers to the query in the way that we can find the required information. Nevertheless, this manner can lead to large results and consequently imprecise results. In other cases, the problem of the queries is coming from the confusion in using terms of the ontology. This leads to the incorrect writing of query with inexistent concepts from the enquired ontology. All these reasons have motivated us to create an extended SPARQL query system based on the procedure of correction and relaxation. It starts by adjusting the query with right terminology from the ontology. This adjustment is made by getting the best matching of the written query with concepts in

the ontology. For that, we have defined specific semantic similarity measure to find the best matching to adjust the query. This procedure of correction can be extended with the relaxation approaches after being sure that the query is semantically correct. Also, we have shown the integration of this new SPARQL querying system into a global architecture of IoT platform. It is established by getting date form IoT devices, creating a description of data coming from, and then incorporating these data into the application information system based on ontology semantic description.

In order to prove the efficiency our contribution, we have evaluated our approach regarding the classical approaches for extending and improving SPARQL querying system. The evaluations are made based on a set of queries applied ontology that describes a case study of IoT applications as the smart Garden Park. They show that our contribution performs better results according to the precision of the extracted information from what the user request. Thus, this extension of SPARQL querying will help the user to write correct and accurate queries.

7. REFERENCES

[1]   M. Hung, "Leading the IoT, Gartner Insights on How to Lead in a Connected World," Gart. Res., pp. 1–29, 2017.

[2]   J. Munir, "State-of-the-art of Internet of Things ontologies," Tech. Univ. Berlin, 2016.

[3]   P.Wongthontham and B. Abu-Salih, "Ontologybased Approach for Semantic Data Extraction from Social Big Data: State-of-the-art and Research Directions," Curtin Univ., no. Ibm 2015, pp. 1–40, 2018.

[4]   A. K. B, Z. Jan, A. Zappa, and M. Serrano, "Overcoming the Heterogeneity in the Internet of Things for Smart Cities," insight-centre, vol. 10218, pp. 20–35, 2017.

[5]   C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big Data Privacy in the Internet of Things Era," IT Prof., vol. 17, no. 3, pp. 32–39, 2014.

[6]   A. Bhadani and D. Jothimani, "Big Data: Challenges, Opportunities and Realities," Eff. Big Data Manag. Oppor. Implement, pp. 1–24, 2016.

[7]   M. Nguyen, M. Lee, S. Oh, and G. Fox, "SPARQL Query Optimization for Structural Indexed RDF Data," Grids.Ucs.Indiana.Edu, 2014.

[8]   Michael Compton et al, "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group", Journal of Web Semantics, 2012.

[9]   A. K. B, Z. Jan, A. Zappa, and M. Serrano, "Interoperability and Open-Source Solutions for the Internet of Things," insight-centre, vol. 10218, pp. 20–35, 2017.

[10] M. Nguyen, M. Lee, S. Oh, and G. Fox, "SPARQL Query Optimization for Structural Indexed RDF Data," Grids.Ucs.Indiana.Edu, 2014.

[11]   A. P. Kumar, A. Kumar, and V. N. Kumar, "A Comprehensive Comparative study of SPARQL and SQL," Int. J. Comput. Sci. Inf. Technol., vol. 2, no. 4, pp. 1706–1710, 2011.

[12] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," Proc. ACL, vol. 1, 1995.

[13] G. Varelas, E. Voutsakis, P. Raftopoulou, E. G. M. Petrakis, and E. E. Milios, "Semantic similarity methods in wordNet and their application to information retrieval on the web," Proc. seventh ACM Int. Work. Web Inf. data Manag. WIDM 05, pp. 10–16, 2005.

[14] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. M. Petrakis, and E. Milios, "Information Retrieval by Semantic Similarity," Int. J. Semant. Web Inf. Syst., 2006.

[15] K. Çelik, "Comprehensive Analysis Of Using WordNet, Part-of-Speechtagging, and Word Sense Disambiguation in text Categorization" Comput. Eng. Bahçeşehir Univ., 2009.

[16] Blanchard, E., Harzallah, M., Kuntz, P.: A generic framework for comparing semantic similarities on a subsumption hierarchy. In proceedings of 18th European Conference on ArtificialIntelligence (ECAI), pp. 20-24, Patrace, Greece 2008.

[17] Blanchard, E., Kuntz, P., Harzallah, M., Briand, H.: A tree-based similarity for evaluating concept proximities in an ontology, in Proceeding of 10th Conf. Int. Federation Classification Soc., pp. 3–11. Springer, 2006.

[18] Kamoun, K.,Ben Yahia, S., : Information content similarity measure to assess stability during ontology enrichment. International Review on Computer and Software (IRECOS), Vol.7, N. 3, Mai 2012.

[19] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," Proc. 21st Natl. Conf. Artif. Intell., vol. 1, pp. 775–780, 2006.

[20] S. Harispe, S. Ranwez, S. Janaqi, and J. Montmain, "Semantic Similarity from Natural Language and Ontology Analysis," cs.AI, 2017.

[21] A. Hliaoutakis, "Semantic Similarity Measures in MeSH Ontology and their application to Information Retrieval on Medline," Interface, pp. 1–79, 2005.

[22] A. M. Collins and E. F. Loftus, "A spreading-activation theory of semantic processing," Psychol. Rev., vol. 82, no. 6, pp. 407–428, 1975.

[23] V. A. A. Ayala, M. Przyjaciel-Zablocki, T. Hornung, A. Schätzle, and G. Lausen, "Extending SPARQL for Recommendations," Proc. Semant. Web Inf. Manag. Semant. Web Inf. Manag., no. JUNE, pp. 1–8, 2014.

[24] G. Fokou, S. Jean, A. Hadjali, and M. Baron, "QaRS: A user-friendly graphical tool for semantic query design and relaxation," EDBT 2015 - 18th Int. Conf. Extending Database Technol. Proc., pp. 553–556, 2015.

[25] Miyuru Dayarathna, Comparing 11 IoT Development Platforms; An easy-to-read table comparing the various features of several popular IoT software platforms, 2016.

[26] P. Barceló, L. Libkin, and J. L. Reutter, "Querying graph patterns," Proc. 30th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst., pp. 199–210, 2011.

[27] K. Kamoun and S. Ben Yahia, "Stability Assess Based on Enhanced Information Content Similarity Measure for Ontology Enrichment," in Model and Data Engineering, 2014, pp. 146–153.

[28] J. Ferlež and M. Gams, "Shortest-path semantic distance measure in WordNet v2.0," Inform., vol. 28, no. 4, pp. 381–386, 2004.

[29] E. Blanchard, M. Harzallah, and P. Kuntz, "A generic framework for comparing semantic similarities on a subsumption hierarchy," 18th Eur. Conf. Artif. Intell., pp. 20–24, 2008.

[30] T. Slimani, "Description and Evaluation of Semantic Similarity Measures Approaches," Int. J. Comput. Appl., vol. 80, no. 10, pp. 25–33, 2013.

[31] P.Neirotti, A.DeMarco, A.Corinna, "Current trends in Smart City initiatives: Some stylised facts", Cities, Elsevier, Volume 38, Pages 25-36, 2014.

[32] H. Huang, C. Liu, and X. Zhou, "Approximating query answering on RDF databases," World Wide Web, vol. 15, no. 1, pp. 89–114, 2012.