Analysis on partial relationships in LOD [1]

Abstract

Relationships play a key role in Semantic Web to connect the dots between entities (concepts or instances) in a way that enables to absorb the real sense of the entities. Some interesting relationships would give proof for the existence of subject and object in triples which in tern can be defined as evidential relationships. Identifying evidential relationships will yield solutions to some existing inference problems and open doors for new applications and research. ***Part_of*** relationships are identified as a special kind of an evidential relationship out of membership, causality and etc. Linked Open data as a global data space would provide a good platform to explore these relationships and solve interesting inference problems. But this is not trivial because LOD does not have a rich schema in terms of the data sets and also the existing work with respect to schema mapping in LOD is limited to concepts and not relationships. This project is based on finding a novel approach to identify partial relationships which is the superset of part_of relationships from LOD instance data by conducting a proper analysis of the data patterns in instance data. Ultimately this approach would provide a way to enhance the shallow schemas in LOD which in tern would be helpful in schema matching in LOD. We apply the determined approach to the DBpedia data set in order to identify the partial relationships in *DBpedia*.

Introduction

The most significant aspect of Semantic Web is how things are related to each other. But while relationships being the most important factor, still there is very few work carried out based on the relationships compared to concepts. Finding facts about relationships by itself can lead to interesting research and development based questions[1]. An evidential relationship is a kind of relationship that would provide evidence to the existence of the subject and/or object. In simple terms: given the property, if subject gives the evidence for the existence of the object then it is identified as an "**evidential relationship**". It identified that these kinds of relationships would be very useful in abductive reasoning.Part_of relationships, membership relationships and causality relationships are kinds of relationships that can be defined as evidential relationships. Out of these relationships the focus of the project is based on identifying **Part_of** relationships.

With the fact that Linked Open Data (LOD) contains lots of publicly available data sets connect to each other; LOD would serve as a good platform to identify part_of relationships. But this became challenging because LOD is not very rich in terms of schema information and some of the relationships even are not defined in the schema and they exist only at the instance level. Also there were no any known mappings from upper level ontologies to get an idea about these kinds of relationships. Even though there are pretty good schema mapping systems for mapping schema information, they only consider the concept mapping and very few work exists related to relationship mapping as well. Another important point to consider is that most of these matching mechanisms are done based on the schema information but obviously with the nature of LOD data sets it is all about instance data and there is no known work to the best of our knowledge

[1] resource persons : Cory Henson & Prateek Jain

that consider the instance data for these kinds of mappings.

Our approach in identifying part-of relationship is based on applying property characteristics in partial order theory[8] to the LOD instance level data with a closer look at the data patterns in LOD. As a practical application, we applied our approach to DBpedia[2] data set which is one of the largest data sets in LOD in order to retrieve the partial relationships.

Background

As mentioned in the introduction as well LOD data sets are not very rich in terms of schema information. Most schemas in LOD data set only provide the class hierarchy and few information about the existing relationships. Most of the relationships are only defined in the instance level but not in the schema level. Some of the data sets do not have any schema information like OpenEI data set, government data set and etc . Lack of schema knowledge would badly affect to the efficient querying on LOD as well. The community already identified this issue and there are some previous work like BLOOMS related to mapping schema information so that it enables to query over the data sets more efficiently [8]. But most of these identified schema matching systems like S-Match[5], Aroma[3] and Alignment API [4] only deal with concepts but not on the relationships even though relationships plays a critical role in querying. There are some ontology matching systems like Agreement Maker [6] which maps the relationship information as well but even AgreementMaker did not work well with respect to LOD schemas. This might be due to the fact that LOD data sets do not provide rich schema information.

The mappings between properties would be more meaningful if it is possible to map schema information with respect to some upper level ontologies so in that case these upper level ontologies can serve as a unifying schema for the data sets. Upper level ontologies like

Proton[3], UMBEL[4], OpenCyc[5], and SUMO[6] are used in LOD to map at the instance level data.

But out of these ontologies it is only Proton that maintains manually mapped upper level mappings to DBpedia schema. This mapping contains 27 mappings at the relationship level but we could identify only two part_of relationships ***location*** and ***foundationPlace.*** FoundationPlace property has many locations in different names.[2]

The absence of a systematic approach to schema mapping with respect to identifying

---

[2] http://dbpedia.org/About
[3] http://proton.semanticweb.org/
[4] http://umbel.org/
[5] http://www.cyc.com/opencyc
[6] http://www.ontologyportal.org/
[7] http://en.wikipedia.org/wiki/Partially_ordered_set

relationship mapping led us to find a systematic approach to identify partial relationships in DBpedia data sets.

Approach

Order theory is a branch of mathematics that studies various kinds of binary relations that capture the intuitive notion of ordering, providing a framework for saying when one thing is "less than" or "precedes" another[7]. It identifies orders as special binary relations. For binary relations R that applies on top of a set is said to be partial order if it is,

- Reflexive R(a, a) for all a
- Asymmetric R(a, b) -> NOT R(b, a)
- Transitive R(a,b) and R(b, c) -> R(a, c)

Our approach uses partial order relationships in order to identify part_of relationship as the initial step, to reduce the search space of the part_of relationship as part_of relationship is a subset of partial order relationships.

With respect to the relationships along with triples people hardly define the reflexive property for the data sets in LOD because it seems to be obvious for anyone creates the data. According to our observations, people are not interested in specifying the reflexive property in other data sets other than LOD. Due to this reason it would be reasonable to look only at the Asymmetric and Transitive property in order to identify the partial order relationships. We made this as a reasonable assumption for the approach we have taken.

Even though we use DBpedia as the testing data set we believe this approach is generic enough to apply for any data set in LOD.

Implementation

The architecture for the implemented system is given in the Figure 1

1. Architecture

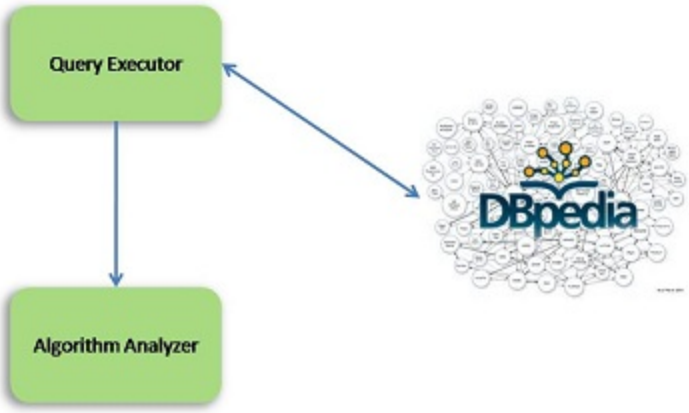

Figure 1: Architecture of the system

The current implementation has two basic parts. For query building, execution, result gathering and analysing results using the algorithm.

2. Query execution

Queries are built for the given set of properties to extract triples satisfying transitive and asymmetric property restrictions. Queries would be executed over the DBpedia sparql endpoint and retrieved results are stored in the JSON format. These sets of triples are analyzed applying the algorithm.

3. Algorithm Search

For the time being, algorithm is set to identify partial relationships. This is the super set containing part of relationships. The logic is as follows. If there exists more than one transitive relationship count and zero symmetric relationship count for a given property then it is identified as a partial relationship. Which in tern will lead to determine part of relationships.

Initial Evaluation

Evaluation is done by comparing the set of relationships acquired by running the algorithm with the manually mapped set of relationships.

Due to practical limitations, exact number of each relationship can not be correctly stated at the moment. The reason for this is that we process all the computations based on our initial result set for each property and do not get dynamic results for transitive and asymmetric queries. We simply process transitive and asymmetric computations on the initial data set without getting a new result set. If a query is executed for each triple, then the whole computation time is exponentially dominated by the time required to fetch each result set from the SPARQL end point. The property list received from DBPedia has datatype properties as well. Hence, the list should be corrected before executing the queries.

Manual mapping had 72 part of relationships. Algorithm retrieved 49 partial relationships. Out of these 49, 16 are part of relationships (37.20%).

Identified problems & examples

location

Manual mapping has this as a “part of” property but the algorithm did not identify as a partial relationship. The reason is that the data set has symmetry in addition to transitivity restriction (algorithm solely depend on the dataset level information).

:North_Main-Bank_Streets_Historic_District :location :Albion_%28village%29,_New_York
:Albion_%28village%29,_New_York :location :North_Main-Bank_Streets_Historic_District

academicAdvisor

Algorithmic search retrieves this as partial relationship but the mannual mapping does not have this. The reason for this appearing in the algorithmic search is that it has the transitivity. But if we consider the meaning of the property this cannot happen. This proves that there exists inconsistencies at instance level in LOD.

Research side effects

We were interested in finding a solution based on dataset level information and not schema information. The main reason for this is the fact that most of the time schema are not complete with required information. This is due to several reasons as schema are developed for a particular requirement in mind and not for every purpose.

In other words, this research can possibly lead to enrich schema information based on dataset evidence. For example, if we identify there exists transitive relationship in data level but schema does not specify, then we can update schemainformation based on new findings. This will be very useful because we will be able to improve our knowledge base in the way of explicitly specifying the restrictions.

This may be an interesting new research idea on improving or enhancing existing ontologies

with unexplored knowledge which is already present in the data set layer.

Future Work

With respect to the future work we would like to identify an approach to distinguish part_of relationships from identified partial relationships and will see the scalability of this piece of work in other data sets as well. We would like to expand this piece of work to identify other interesting relationships like membership and causality relationship as well.

Timeline

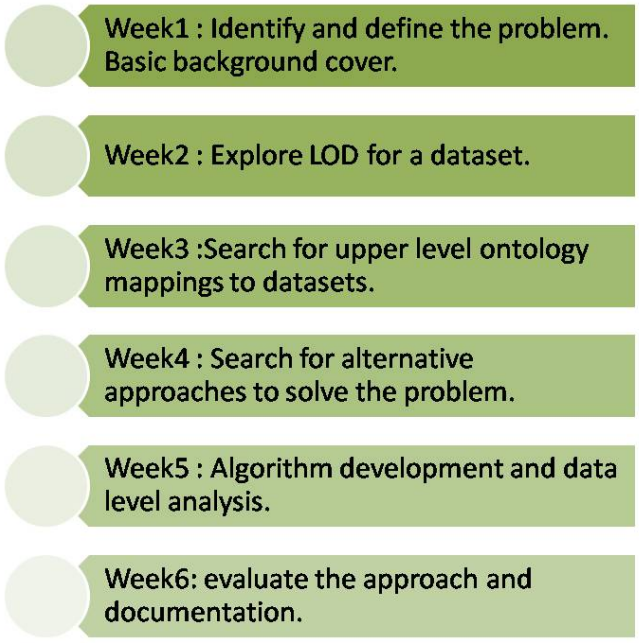

Figure 2: Timeline

Project timeline is shown in Figure 2. The timeline reflects that most of the time is spent on exploring LOD datasets to find a way of identifying relationships.

Work division

- Analyzing different LOD data sets and schemas – Kalpa, Sarasi
- Analyzing Ontology matching systems - Sarasi
- Analyzing Upper level ontology – Kalpa, Sarasi
- Implementing the algorithm
    - Instance data analysis – Kalpa
    - DBPedia mailing list and contacts – Sarasi
    - Partial order feasibility – Kalpa
    - Querying logic to retrieve required data – Sarasi
    - Algorithm logic - Kalpa

References

[1] Amit Sheth, I. Budak Arpinar, V. Kashyap. (2004). Relationships at the Heart of Semantic Web: Modeling, Discovering, and Exploiting Complex Semantic Relationships, in Nikravesh et al, Eds, Enhancing the Power of the Internet (Studies in Fuzziness and Soft Computing) V.139. (pp. 63-94). SpringerVerlag

[2] Damova, M., Kiryakov, A., Simov, K., Petrov, S.: Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I., eds.: In Proceedings of the Fifth International Workshop on Ontology Matching. CEUR Workshop Proceedings (November 2010)

[3] David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM 2006, pp. 830–831. ACM, New York (2006)

[4] Euzenat, J.: An API for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)

[5] Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an algorithm and an implementation of semantic matching. In: Kalfoglou, Y., et al. (eds.) Semantic Interoperability and Integration. Number 04391 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany (2005)

[6] Isabel F. Cruz , William Sunna , Anjli Chaudhry, Ontology alignment for real-world applications, Proceedings of the 2004 annual national conference on Digital government research, p.1-2, May 24-26, 2004, Seattle, WA

[7] Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In Patel-Schneider, P., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J., Horrocks, I., Glimm, B., eds.: Proceedings of the 9th International Semantic Web Conference, ISWC 2010