Alignment and Dataset Identification of Linked Data in Semantic Web *

Kalpa Gunaratna, Sarasi Lalithsena and Amit Sheth
Kno.e.sis Center, Wright State University, Dayton, OH 45435 USA
{kalpa, sarasi, amit}@knoesis.org

Abstract

The Linked Open Data (LOD) cloud has gained significant attention in the Semantic Web community over the past few years. With rapid expansion in size and diversity, it consists of over 800 interlinked datasets with over 60 billion triples. These datasets encapsulate structured data and knowledge spanning over varied domains such as entertainment, life sciences, publications, geography, and government. Applications can take advantage of this by using the knowledge distributed over the interconnected datasets, which is not realistic to find in a single place elsewhere. However, two of the key obstacles in using the LOD cloud are the limited support for data integration tasks over concepts, instances, and properties, and relevant data source selection for querying over multiple datasets. We review, in brief, some of the important and interesting technical approaches found in the literature that address these two issues. We observe that the general purpose alignment techniques developed outside the LOD context fall short in meeting the heterogeneous data representation of LOD. Therefore, an LOD specific review of these techniques (especially for alignment) is important to the community. The topics covered and discussed in this article fall under two broad categories, namely alignment techniques for LOD datasets and relevant data source selection in the context of query processing over LOD datasets.

Introduction

Sir Tim Berners-Lee introduced the idea of Linked Data based on four simple rules[1] to publish RDF[2] based datasets on the Web. The four founding rules are: (1) use URIs (Uniform Resource Identifiers) for naming things, (2) offer the ability to look up URIs, (3) provide useful information upon URI lookup, and (4) include links to other URIs. This set of simple rules laid the foundation for creating the "Web of Data", which is a collection of interlinked datasets (also termed "Linked Open Data"). Currently, the Linked Open Data (LOD) cloud consists of over 800 datasets[3] covering numerous domains like entertainment, life sciences, government, publication, events, etc. Bizer et al.[4] pointed out how the four founding principles have evolved to create the LOD cloud with RDF datasets and how these interlinked datasets could be used in applications. The rapid growth in publishing interlinked datasets on LOD by various communities over the past five years made the LOD cloud an experimental platform for interesting applications such as knowledge discovery and question answering[5, 6]. In this regard, the Linked Data concept has taken a big leap in the technology and vision of Semantic Web.

As the LOD cloud continues to rapidly develop towards serving the above mentioned interesting applications, it brings forth new challenges in data integration, relevant data source identification, query formulation, etc. Data integration over LOD becomes inevitable and beneficial since interconnected datasets often have complementary data. Hence, a unified integrated view of the

*Pre-print version of the published article. DOI: 10.1002/widm.1121

facts of a concept, which reside over several datasets, can produce a complete picture of the concept spanning over different viewpoints. Moreover, data integration on these interlinked datasets requires alignment techniques over different granularities, as concept and property in the schema level and instance in the data level. These alignment techniques not only service data integration tasks, but also exploration and querying LOD as a whole, as they make up connections on LOD at both schema and data (instance) levels. The growth in the number of datasets brings forth challenges in identifying relevant datasets that could be matched for a given task, as it is impossible to lookup relevant information in each dataset individually. Furthermore, the relevant data source selection problem has garnered high levels of interest to the linked data query processing community because it directly affects the execution of an efficient query plan. An example query expressed in natural language over LOD datasets, resembling these issues can be outlined and explained as follows.

"*Identify Congress members who have lived in Capitol Hill for the past four years, who also have mines or power plants in their congressional districts.*"

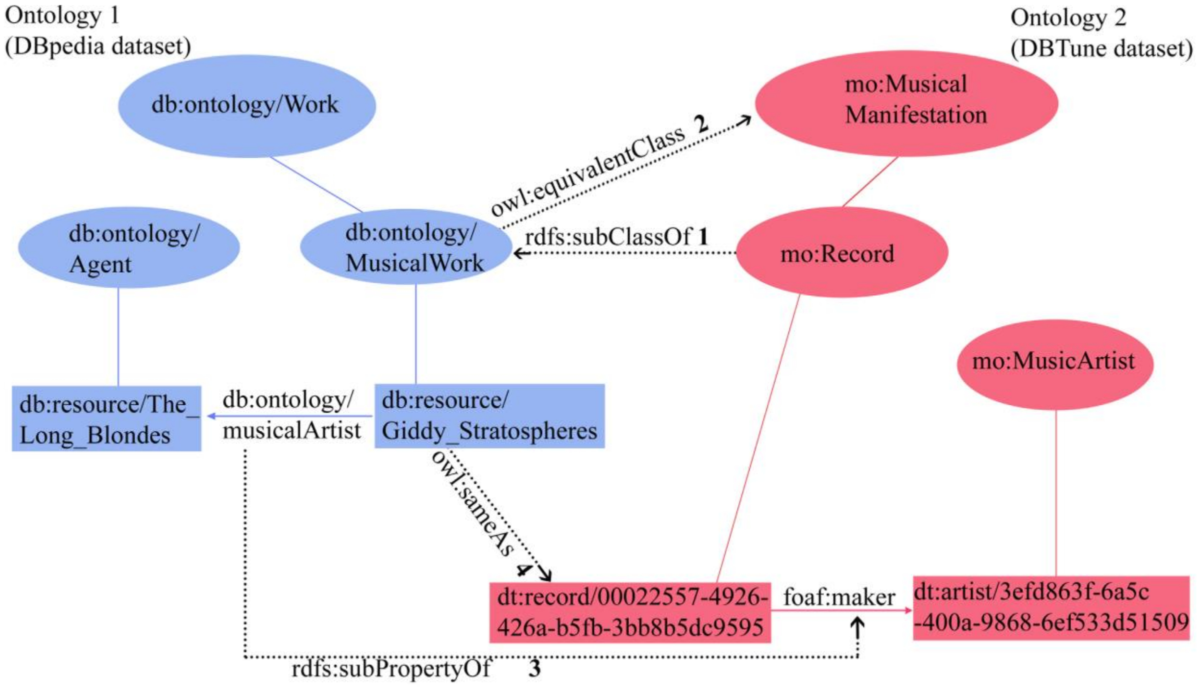
Answering this query requires searching for facts in multiple datasets. DBpedia[7] and GovTrack[8] datasets have "congress member" details and member "time periods" can be found in the GovTrack dataset. Locations of "mines" and "power plants" are in the Geonames[9] dataset, whereas the relevant "congressional districts" are in the US Census[10] dataset. Getting similar information from two datasets as in DBpedia and GovTrack needs alignment techniques and to identify the above mentioned datasets over many other datasets requires relevant data source selection.

Therefore, addressing both alignment and data source selection problems is imperative in the LOD context. Alignment techniques create various links between datasets and build the vast data space of the interconnected Web of Data. Locating relevant sources becomes challenging when there are a large number of individual datasets in this data space with overlapping and complementary information. The first part of the article discusses different approaches for ontology alignment in the levels of concept, property, and instance, whereas the second part discusses systems and their techniques in identifying relevant data sources for query processing in LOD. We conclude with possible interactions between these two areas in brief.

1. Ontology Alignment in Linked Open Data

An Ontology is "an explicit specification of a conceptualization"[11], a definition introduced by Thomas Gruber. Willem Borst extended this definition as he thought the original definition was too broad. He argued that there should be agreement on the conceptualization because an ontology will not be re-usable if it is not generally accepted. Borst presented his definition of an ontology as "a formal specification of a shared conceptualization"[12]. Along with this definition, others have identified that sharing knowledge and structure is one of the many important contributions of an ontology[13,14].

Data instances can be linked to these concept hierarchies (populating ontology), and relationships (also termed "properties" or "predicates") among these instances can be defined. The ontology level information regarding data definitions, relationships, rules, etc. is known as schema information, whereas data instance information representation is known as the instance/data level. “*Ontology Alignment”* in general is about finding alignments (or correspondences) between concepts, properties, or instances in two or more ontologies based on their similarities (see Figure 1). In the LOD context, it mainly comprises three parts: concept (class) level alignment, property alignment, and instance alignment (interlinking or entity co-reference). The first two are about the schema level agreements while the latter is about data level agreements. Datasets in LOD are for the most part linked to each other by instance level relationships (*owl:sameAs, skos:exactMatch, etc.*), which are created using instance alignment[15-17] techniques, but similar relationships between concepts and properties are not inherent. Concept alignment in LOD has been investigated to some extent[18-21] in the recent past and showed significant progress both in precision and coverage, but property level alignment is yet to achieve considerable attention, coverage, and results other than a recent effort by Gunaratna et al.[22].



namespaces:
db -> http://dbpedia.org/, dt -> http://dbtune.org/musicbrainz/resource/, mo ->http://purl.org/ontology/mo/, foaf -> http://xmlns.com/foaf/0.1/

Figure 1 Concept, property, and instance alignment example

The three types of alignments can be explained using a sub-section of two ontologies (DBpedia ontology and Music ontology[23]) found in the LOD cloud as shown in Figure 1. It illustrates how concept, property, and instance alignments can be stated between two datasets having example matches for each alignment type. The two datasets (DBpedia and DBTune MusicBrainz[24]) model knowledge on music in different viewpoints for two specific instances, but they both have similarities, which can be matched/aligned. Parts of ontology 1 (DBpedia ontology) and ontology 2 (Music ontology) are drawn in blue and red respectively, and the figure contains two instances for each dataset populated using the ontologies. The concepts are shown in oval shapes, whereas the instances are shown in rectangles. The example alignments between these two ontologies are marked using black dotted lines numbered from 1 to 4. Number 1 and 2 represent an example of concept alignment showing equivalent and sub-class relationships found between datasets. Number 3 and 4 show property and instance alignment examples found respectively (note: the alignments are not within the same dataset).

1.1 Concept Alignment

Concept alignment techniques on LOD can be categorised into two main broad categories as systems using: (1) external hierarchies and knowledge present in lexical databases like WordNet[25] and online encyclopedias like Wikipedia[26], use of Natural Language Processing (NLP) techniques, and (2) instance level information. The classification of systems presented in this article is based on the type of systems available on the LOD setting and a more general and comprehensive listing of techniques could be found in[27]. Often, similar or related concepts in two ontologies do not have simple synonymous interpretations and hence simple synonym based approaches do not have significant coverage in LOD. Therefore, state of the art concept or schema alignment techniques[28, 29] (ones that also performed well in OAEI[30] - Ontology Alignment Evaluation Initiative) often cannot be used without making significant changes in the LOD setting.

1.1.1 Use of external hierarchies and NLP techniques

In searching for a solution for the alignment challenge described above, Jain et al.[18] proposed a bootstrapping based system called BLOOMS. BLOOMS builds upon the idea of using external community built concept hierarchies for the alignment process. Wikipedia is a free and high quality encyclopedia, which is continually maintained by the open community where each page is categorized under a set of topics. BLOOMS explores this category hierarchy for concepts to be aligned between ontologies and builds a set of tree data structures (forest) for each concept. Then, calculating the overlap $o(T_s, T_t)$ of trees from two forests for two concepts (s and t) yields a measurement for similarity. If the overlap is equal to the number of nodes in each tree, the concepts are considered equivalent, otherwise a sub-class relationship is determined. That is, concept *s* is a sub-class of the other concept *t* if the overlap value of *s* is less than the overlap value of *t.* The overlap is calculated using the number of shared terms over the total number of terms in the trees. Gruetze et al.[31] incorporated the idea of BLOOMS forest construction to compute mappings between ontology concepts to an approach called Holistic Concept Matching. The idea of the approach is to minimize the number of concept pair comparisons by grouping concepts according to topics. The topic sets for concepts are determined by ranking Wikipedia forest tree nodes using tf-idf measurement. Then these topic sets are analysed for aligning concepts.

BLOOMS was also evaluated as a general purpose ontology alignment system with the other existing ontology alignment systems like AROMA[32], RiMoM[33], and S-Match[34] outside the LOD domain in[18]. S-Match uses an approach of semantic matching by understanding the semantic meaning codified implicitly or explicitly in the labels. Furthermore, S-Match uses string manipulation for weak semantic matchers and WordNet for its strong semantic matchers. RiMoM quantitatively estimates textual and structural characteristics and uses them accordingly for the alignment. AROMA on the other hand utilizes an association rule mining concept, which is frequently used in the database domain. BLOOMS was shown to be very competitive among all these existing general purpose ontology alignment systems and outperformed them in many cases because of its diverse hierarchical structural mapping ability and coverage using Wikipedia. When considering ontology alignment in LOD, the general purpose ontology alignment systems are particularly challenged because of the multi-domain coverage of LOD. A system such as BLOOMS that uses broad background knowledge will likely produce better overall precision and recall when facing this challenge. BLOOMS+[19] is an enhanced version of the BLOOMS system where it addresses some of the shortcomings by taking into account the size of trees in logarithmic scale and penalizing matching nodes appearing in the deeper parts of the trees since those concepts seem to be more generic and could add noise to the matching process. Furthermore, BLOOMS+ compares the super-category of each concept to match the context. For example, it is able to identify "Jaguar" and "Cat" as, not a possible alignment considering the fact that "Jaguar" has a super category "Car" and "Cat" has a super category "Mammal" whereas in BLOOMS it could have just identified Jaguar as a mammal and not a car type. In this regard, BLOOMS+ has improved the BLOOMS framework significantly for LOD ontology alignment tasks and evaluated its claims using manual mappings of concepts in DBpedia, Geonames and Freebase[35] to Proton[36]. Proton is an upper level ontology, which consists of about 300 classes (concepts) and 100 properties, providing coverage for general concepts for a wide range of tasks including semantic annotation, indexing, and retrieval of documents.

AgreementMaker[37] is considered to be an efficient ontology/schema alignment system in the classical setting, one that OAEI evaluation represents. Cruz et al. have adapted AgreementMaker to implement an efficient system called "OnTheGO matching of Linked Open Data ontologies"[38] to align LOD ontologies. One of the primary goals of this system is to avoid long processing times encountered by BLOOMS-like systems for computing similarities in tree/forest data structures. The system implemented two methods to discover a mapping between two ontologies based on similarity metrics and a third party ontology to discover equivalent, sub-super class relations between concepts. The third party ontology (mediator ontology) in this case is WordNet. The system is compared with AROMA, S-Match, and BLOOMS and the average results are competitive, while BLOOMS has better recall values.

1.1.2 Use of instance level information

The idea of utilizing instance data for concept alignment has also been considered in the recent past and shown to be effective[20, 21, 39, 40]. Parundekar et al.[39] proposed that identifying equivalent instances belonging to concepts leads to an alignment between those concepts. To identify equivalent instances, they utilise properties-like *owl:sameAs*, *skos:closeMatch*, etc. that link instances across datasets. Even though there are issues related to whether *owl:sameAs* links exactly the same instances[41], such experiments demonstrate its applicability in general. As a follow up, Parundekar et al.[20] developed an alignment technique based on concept coverings, which aligns concepts as well as aids in curating linked datasets for missing or incorrect data. Findings in[20, 39] showed how the technique can be of benefit especially in areas where concepts are vague and tools such as BLOOMS and AgreementMaker can fail. Moreover, the system is able to find one-to-one and composite (a set of classes making a concept/class) concept coverings. Along this line of work, Correndo et al.[21] incorporated a statistical approach utilizing *owl:sameAs* links between instances with the Jaccard co-efficient measurement to measure the overlap of instances in aligning concepts. Nikolov et al.[40] also utilized *owl:sameAs* links to infer mappings between ontology concepts in the

LOD. They trained a classifier based on instance overlaps for concepts to determine the mappings. These mappings can be of low quality as the mapping is based on strong degree of instance overlap but serves the system's intended purpose of recommending to the user other available related concepts in the LOD. PARIS[42] is a probabilistic alignment approach that can be applied to concepts, instances, and properties. PARIS utilizes the instance overlap to compute the subclass relationships between the concepts. Such systems attempt to utilise the inherent linking nature on the instance level of LOD for the aligning process, thus taking alignment research into new directions.

1.2 Property Alignment

Property alignment presents an important and complicated alignment challenge in the ontology alignment field. This is because properties capture complex structure and meaning of the instance level data, whereas classes have more abstract meaning. In spite of its importance, research and tools have been not given the level of attention it deserves in the LOD domain. Some techniques have been proposed based on similarity metrics, clustering, machine learning, and more recently using property extension matching. Property alignment has two components: data-type and object-type property alignments. Object-type properties are the ones having RDF resources as both subject and object of the property. Data-type property alignment is primarily centered on string similarity based metrics. Tran et al.[43] proposed a cluster based technique using four similarity metrics such as string similarity, WordNet similarity, profile similarity, and instance similarity for ontology alignment. The system uses the same technique used for concept alignment for properties, which is based on weighted similarity measures. The results demonstrated on OAEI benchmarks were not competitive and need further refinements to improve performance. Sleeman et al.[44] incorporated a density estimation approach using Kernel Density Estimation (KDE) to map opaque properties. Opaque properties are properties conveying the same meaning, having similar names or different names. The proposed technique can be applied to both types of properties but the need of transformation of values into a numerical format, to be compatible with KDE is problematic. This transformation can be difficult in the LOD domain.

Graph based ontology analysis and learning proposed by Zhao et.al[45] is an approach for querying linked datasets by developing an upper level ontology using ontology learning techniques. They use a property grouping strategy for aggregating similar properties based on object overlap (in triples) found in the datasets. But the approach is not suitable for finding property mappings since it can group semantically different properties like “birthPlace” and “deathPlace” into one group. TripleRank[46] is a system built for faceted browsing over linked data and as a by-product of the Singular Value Decomposition (SVD) process, it claims to identify equivalent properties within a dataset. However, no evaluation is available to show to what extent it can handle identifying equivalent properties among datasets.

Gunaratna et al.[22, 47] proposed a successful approach that can be used in the LOD environment by utilizing existing links between the data instances to match property extensions for property alignment. These links are the Entity Co-Reference (ECR) links, which are used to link two semantically same instances in two datasets. Property extension for a property *P* in a dataset is defined as the set of all the Subject (*S*) and Object (*O*) pairs (*S,O*) that the property is connected to, in the dataset. The idea behind the approach is that semantically same properties in two datasets have more matching subject-object (S,O) pairs in their property extension. There arises the issue of coincidental matches as for example "birthPlace" and "deathPlace" properties, where many matching (S,O) pairs can be found when many people are born and dead at the same city. But when analysing the aggregated results for a larger sample, these coincidental matches can be eliminated. This elimination of incorrect matches was handled by using several statistical measures found in the extension matching. One of the limitations of the approach is that it uses ECR links in the matching process and when the ECR links are sparse, the matching process cannot be performed successfully. The other is that, it may be used with overlapping datasets where common facts and entities are present whereas totally different datasets in the same domain may not produce results. The results of the alignment shows better alignment ability over current syntactic and WordNet based approaches for the LOD cloud. Along this line of work, Zhang et al.[48] proposed the concept of Statistical Knowledge Patterns (SKP) to cluster synonymous property pairs and tested it with the DBpedia dataset. They analysed subject overlap, triple overlap (subject and object overlap for two properties) and cardinality of the properties to define property similarity in using agglomerative clustering techniques but limited to intra-dataset analysis.

The general topic of property alignment for ontologies has been addressed by the above systems, but none of them are tested in the LOD context except for the extension based approach proposed by Gunaratna et al.[22]. Lack of approaches proposed and techniques tested on LOD signifies a considerable gap in this area of research but could improve on novel approaches like analysing property extensions together with syntactic and external dictionary based techniques in the future.

1.3 Instance Alignment

Alignment on this level is important because identifying the same entity in different datasets improves data interoperability. The links created in this process are mainly *owl:sameAs* links that intend to link the same entities (by entity we mean the real world object, whereas different instantiations of this entity are instances) and there seems to be other types of links resembling different levels of similarity such as *rdfs:seeAlso*, *skos:closeMatch*, *skos:relatedMatch*. Moreover, as pointed out by Halpin et al.[41, 49], the *owl:sameAs* links are sometimes misused in the LOD context. Often, what they link is not entirely incorrect, but instances with different granularities (i.e., London vs. Greater London). However using *owl:sameAs* to link two similar instances leads to a key question, whether there should be exactly the same thing with two URIs. Finding the *owl:sameAs* semantics between instances in different datasets is defined as instance alignment, interlinking, link discovery, or entity co-reference. The systems for interlinking can be categorized into two parts, as systems: (1) requiring manual link specification including semi-automatic matching, and (2) that automatically identify specifications and domains for interlinking.

Finding similar instances in different datasets is challenging for reasons such as: (1) millions of instances need to be compared with each other that in turn requires a good blocking mechanism (blocking is pruning possible instances, which are irrelevant before comparing pairs for similarity), (2) most instances seem to be matching but are actually different, reflecting the need to have high precision, and (3) many parameters to check as many dimensions available in the instance level. Because of the complex nature of the problem, early attempts for interlinking were manual.

1.3.1 Systems requiring manual link specifications

To make progress towards partial automation, Volz et al.[16, 50] proposed the SILK framework to identify the same entities in different LOD datasets. The system uses a link specification language called *Silk Link Specification Language* (SILK-LSL) to express rules for the matching process to decide the relationship between entities. It makes use of several similarity metrics (string similarity, qgram, taxonomic similarity, etc.) for similarity calculation, which ranges between 0 and 1. These similarity values are then aggregated by several defined functions to decide the best match over a threshold, while the rest that fall below the threshold are manually verified. LIMES[15] is another link discovery system developed to consider the efficiency of calculating the mappings and the volume of data to be processed. LIMES uses the triangle inequality in metric spaces for calculating instance similarities and outperformed SILK showing lesser computation times for large datasets. Based on these triangle inequality measures, LIMES can filter out many instance pairs that cannot suffice matching conditions. Compared to LIMES, SILK uses instance pre-matching, which also causes recall values not guaranteed to be 1. To avoid this problem, SILK adopted the MultiBlock (multidimensional blocking) approach in pre-processing that guaranteed lossless recall[51].

1.3.2 Automatic interlinking systems

1.3.2.1 Unsupervised approaches

SERIMI[17] is a link discovery tool, which consists of two phases. In the first phase, it utilizes traditional information retrieval strategies to select which candidate instances to be aligned. For this, entity labels of the source dataset are used to search for candidate entities in the target dataset. When candidates are selected, they are disambiguated for correctness in the second phase. SERIMI does not require any alignment between ontologies for the process and hence it is able to link instances belonging to the same entity representing different factual representations in two datasets (for example a city expressed using social aspect and geological aspect in two datasets). Song et al.[52] introduced a different approach to the interlinking problem by understanding coverage and discriminability of properties of instances. For example, an instance having a property connected to a rare value (discriminating factor) could lead to a better blocking mechanism and disambiguate the instance from others.

1.3.2.2 Supervised and genetic algorithm based approaches

There are approaches designed to aid interlinking systems by learning rules. They utilize genetic algorithms, supervised, and active learning techniques. EAGLE[53] is a system proposed by Ngonga et al. that utilizes genetic algorithms and active learning techniques that require minimal user interaction in labelling instance pairs for automatically learning link specifications for the matching process. Ngonga et al. further improved the EAGLE system that they call COALA[54]. COALA is an improvement over EAGEL in terms of accuracy and efficiency where active learning is incorporated with correlations of classes. The approach tries to solve the fundamental problem of a user having to provide a link specification for instance matching in systems such as LIMES and SILK by learning the specification by itself. Along with LIMES and EAGLE, Ngonga et al. further extended the work and addressed the theoretical quality in the link discovery framework[55]. Isele et al.[56] introduced a genetic algorithm based approach to the SILK framework to identify link specification rules for instance alignment. They further improved the system to incorporate the active learning paradigm to learn linkage rules in the SILK framework[57].

1.4 Alignments, applications and summary

The practical use of the alignments on LOD can be seen in applications that try to make sense of this immense data. For example, ALOQUS[58] is an alignment based querying system for LOD, built using Proton upper level ontology and its concept mappings to other datasets using BLOOMS[18]. Graph based ontology analysis approach[45] is another kind of approach, it groups concepts and properties on several ontologies to build an upper ontology for querying underlying data. Furthermore, instance alignments are used for both querying and concept alignments[20, 21, 39]. In this sense, the three types of alignments we briefly discussed are tightly coupled with interesting applications as well as among themselves. Therefore, it is important to have links not only in the data level but also in the schema level as well. Even though property alignment takes an important place in data integration and organization of the integrated results, it is yet to achieve its maturity. Concept and instance alignments have shown considerable progress over the past years but could be further improved for higher precision and recall values. Furthermore, to gain the full potential of this large set of datasets, many other useful relationship types such as partonomy, which is to some extent explored by Jain et al.[59] and causality, which is hard to capture, should be investigated. Hence, in the future, the research community will need to look at important types of complex relationships, such as partonomy and causality, and understand how such relationships can be modelled, discovered, extracted, reconciled, and exploited for deeper insights and decision making as in[60] using LOD.

2. Data Source Selection for Querying on LOD

The increasing attention from the diverse range of communities to publish the data and create SPARQL[61] endpoints to access these published data make LOD a good querying platform for knowledge exploration and discovery. Data publishers can easily use a number of available tools and techniques to convert various structured data formats to RDF and make them available for access through SPARQL endpoints. The LOD cloud allows data publishers to publish their data on the web and link with other related datasets giving them more flexibility, avoiding global constraints such as a central schema or choice of word selection. This flexibility for data publication over LOD raises issues for having an overall knowledge about the datasets found in this global space, which is crucial for data consumption. In fact, this directly affects the relevant data source selection for various tasks and applications such as query processing and interlinking. The problem becomes even more challenging with the increasing number of datasets and dynamic nature in terms of adding or removing datasets, and updating their content.

The topic of query processing over datasets has discussed the data source selection problem in detail, considering it as one of the major challenges. For instance, Hartig et al.[62] pointed out that data source selection poses new challenges for query processing on LOD, which is not investigated by traditional federation. Ladwig et al.[63] categorized three state of the art strategies for Linked Data query processing and the categorization is primarily based on the variations of data source selection approaches by different query processing systems. We use the same three strategies to describe the various source selection approaches including approaches used in query processing. The three strategies are,

- Top-Down

  Top-down strategy identifies the relevant data sources using some form of source selection indexes by processing datasets in advance. Data selection approaches use this as a prior knowledge to select the relevant data sources.

- Bottom-Up

  Bottom-up strategy discovers the relevant data sources on the fly by using some form of an input (input in the forms of urls, labels, etc.) as seeds. This strategy does not rely on any prior knowledge about the datasets.

- Mixed strategy

  Mixed strategy uses both top-down and bottom-up approaches to discover relevant data sources appropriately.

In the following sections (2.1 to 2.4), we discuss different source selection approaches that fall under the aforementioned three strategies in the context of federated querying and interlinking.

While querying applications are looking for the datasets which contain the relevant results for a given query, it is also useful to manually identify the relevant data sources for a given task at hand. Existing catalogues such as LOD bubble diagram[64], CKAN[65], and LODStats[3] provide an interface for this purpose by being the entry points for LOD datasets. In section 2.5, we briefly discuss the recent developments on manual selection of the datasets.

2.1 Top-Down Strategy

The top-down strategy mainly relies on various kinds of indexing mechanisms to find relevant data sources. Harth et al.[66] proposed an indexing structure to store the summaries of datasets and leverage this indexing structure to identify the relevant datasets for query processing. The index structure focuses on storing only an approximation of the dataset rather than keeping every entity in the index. The indexing is handled by converting RDF triples into a numerical format using a hash function and index it in the mapping bucket of a "QTree"[67]. QTree is a multidimensional indexing structure, and in this case coordinates are obtained by applying the hash functions to the Subject (S), Predicate (P), and Object (O) of the triples and a bucket contains data items with similar hash values. Once they have a query looking for datasets, it is converted into a numerical format using the same hash function that was used for triple conversion and identifies the matching region from the QTree.

The follow on approach (named SPLENDID) by Görlitz et al.[68] incorporated existing metadata descriptions to build an index, which consists of relevant information for data source selection used by the query federation. SPLENDID[68] uses VoID[69] descriptions for query federation. A VoID description of a dataset has metadata about the dataset such as types, predicates, SPARQL endpoint, and number of triples. SPLENDID collects the statistical information from VoID descriptions and creates a local index, which maps predicates and types to datasets and other statistical information. When executing the query, it assigns datasets for each triple pattern based on mapping bounded predicates and type information in the query with the local index. Whenever there are no bounded predicates in the triple patterns, a SPARQL ASK query is sent to all the collected SPARQL end points to see whether there exist any results for the specific patterns.

FEDX[70, 71] is another query processing system, which follows a top-down strategy for relevant source selection. FEDX issues SPARQL ASK queries for each triple pattern of the query to each SPARQL endpoint (the list of SPARQL endpoints are known in advance) before query optimization. The result of the ASK query is maintained for any upcoming queries with similar triple patterns. But this will overestimate the relevance of a dataset if there is a generic triple pattern such as "?s rdf:type ?o".

SchemeX[72] uses a scalable index structure for indexing LOD datasets, which can be useful in data source identification. Its index structure abstracts RDF instances to classes and builds type clusters based on the identified classes. These type clusters can be further partitioned based on the same outgoing properties for instances of the type clusters. It keeps track of the dataset details along with the type and property information of the dataset. This supports the building of an index without a persistent storage of data by using a stream-based approach.

Even though top-down strategy can identify the relevant data sources with a fast response time by using the prior knowledge stored in the form of an index, it suffers from identifying fresh or more up to date datasets since the results are based on the information collected at indexing time.

2.2 Bottom-Up Strategy

The bottom-up strategy focuses on finding relevant data sources on the fly. Hartig et al.[62] find the relevant datasets on the fly through link traversal techniques. They make use of the de-referenceable nature of URIs, and most importantly the approach does not rely on any indexing mechanisms. Initially they execute parts of the SPARQL query by looking up URIs in the query and then further leverage the other URIs retrieved from the partial results. But in this approach, in order to initiate the query execution it must have initial URIs and at the same time it is possible that the approach fails to retrieve the complete result at the end. Furthermore, the solution can lead to infinite link discovery, where the system is unable to fulfil termination conditions and continues searching for links.

Feedback[73] proposed another approach to data source selection, which also starts with URIs in the application queries to track the relevant datasets. The system crawls datasets by taking these URIs as the seed resources and then looks for other URIs using predicates like *rdfs:seeAlso*, *owl:sameAs,* and *owl:equivalentClass.* After identifying these datasets, the system ranks datasets by analysing user feedback.

Nikolov et al.[74] addressed the relevant data source selection in the context of identifying suitable datasets for interlinking for a given dataset. They extract a sample set of instance labels from the dataset to be interlinked and query those instance labels in Sigma[75] to identify the relevant data sources and then rank those datasets based on the degree of similarity.

Unlike the top-down strategy, bottom-up strategy has the capability to identify more recent (fresh) results, but this may lead to issues like infinite link discovery and slower query time compared to the top-down strategy.

2.3 Mixed Strategy

The mixed strategy tries to get the best from both top-down and bottom-up approaches in order to make sure it retrieves more recent/up-to date results with a fast respond time. This assumes a partial prior knowledge of relevant datasets and further updates knowledge at the time of query processing. Query processing systems described by Ladwig et al.[63, 76] and Umbrich et al.[76] use this approach for source selection. Ladwig et al.[63] use local indexes along with query triple patterns to identify the data sources as an initial list of possible relevant sources and further discover sources based on the content processed from the initial relevant source and intermediate results. The process of finding the relevant datasets terminates based on the preconfigured values such as number of results to produce and number of source datasets. They introduce a ranking mechanism for the sources whenever appropriate to rank more relevant data sources. Ranking is performed by using certain metrics, which use a number of features such as the cardinality (number of triples in a dataset matches with a given triple pattern), specificity (number of constants in a given query triple pattern), and number of incoming links from a relevant resource.

Umbrich et al.[76] proposed a hybrid query plan execution strategy to identify the relevant sources either from materialized indexes (results from the top-down approach) or on the fly queries at run time (results from the bottom-up approach). It tries to identify which strategy can be used to retrieve the results for parts of a query based on statistics and these statistics are based on dynamicity and coverage of materialized indexes.

2.4 The three approaches

The top-down approach relies on having prior knowledge of datasets, which is stored using index like data structures and therefore can be optimized for a fast response time in identifying relevant data sources. But the top-down approach may fail to recognize up-to-date results because the identified datasets are collected at indexing time and the results might be different in querying time. In contrast, the bottom-up strategy finds relevant datasets on the fly during the querying time, which enables identifying up-to-date/fresh results. However, this encounters slow response times compared to the top-down approach. The mixed strategy combines both approaches hoping to maintain a balance between up-to date results and a fast response time. Even though the source selection is discussed with querying applications in detail, it needs to be further improved in the context of source selection for applications such as interlinking.

2.5 Dataset Catalogues for manual data selection

There are well known datasets such as DBpedia[7], Freebase[35], and MusicBrainz[77], and datasets that are not widely known such as ClimbData[78] and Lingvoj[79] that might be useful for certain use cases. It is extremely difficult to identify the potential datasets for a given task without a catalogue of datasets. Existing catalogues such as CKAN and LODStats allow users to search for datasets using keywords, manually assigned tags, and other kinds of metadata. CKAN encourages data publishers to manually tag datasets from a predefined set of tags and use these tags to organize the LOD cloud bubble diagram. LODStats uses a stream-based approach for gathering statistics of the datasets based on the classes, properties, and vocabularies used in the datasets.

While the existing catalogues rely on keywords, manually assigned tags, and known URIs of the datasets, there are some recent approaches proposed to improve the descriptions of these datasets. The improved descriptions (including metadata) of the datasets can be used to better organize this huge data cloud in order to ease the trouble encountered in finding datasets. Frosterus et al.[80] presented a system to create and enrich such metadata about the datasets via annotation tools and faceted search. However, this approach expects that the data publishers or some third party provide the annotations. Lalithsena et al.[81] proposed an approach to automatically identify the domains of these datasets by utilizing Freebase, both as the background knowledge and the vocabulary (Freebase domains and categories). This approach can be used to address the scalability issues in manual tagging of datasets of the aforementioned approaches. Even though this approach provides the ability to automatically identify the topics of the datasets, the topics are limited to the Freebase vocabulary. This work can be useful to categorize the datasets automatically with improved domain coverage. In conclusion, LOD datasets still need efficient mechanisms to catalogue the datasets to identify the relevant data sources.

Conclusion

Ontology alignment and data source selection are considered to be two of the more important research problems among the LOD community over the past few years, because, they can make facts and information present in LOD datasets more useful by providing solutions in tasks like data integration for more complete knowledge acquisition, querying data in finding answers, etc. In this article, we have discussed these two problems highlighting some of the existing systems that attempt to solve them, varying from NLP to information retrieval and background knowledge based approaches. The nature of LOD is such that its knowledge is distributed among many datasets and aligning and querying brings useful information, which cannot be realistically stored in a single place.

Alignment techniques over datasets support merging them together to help fetch information in querying and most importantly make up the LOD cloud by creating connections in both schema and data levels. However the merging of all possibilities is not a viable solution unless the relevant data sources are identified. Hence, identifying which datasets to align and query is also equally important. Therefore, techniques developed in alignment and source selection will indeed make steps towards realizing the potentials of these huge interconnected datasets (in a sense, knowledge bases). In conclusion, LOD contains many datasets covering many domains and consuming this vast knowledge requires alignment and identification of relevant data sources. The article reviews these issues and solutions highlighting the need for LOD specific techniques in using the LOD cloud for applications and future research directions.

References

1. Berners-Lee T. Linked data-design issues (2006). Available at: http://www.w3.org/DesignIssues/LinkedData.html. (Accessed 23/03/2013)
2. Klyne G, Carroll JJ, McBride B. Resource description framework (RDF): Concepts and abstract syntax. Available at: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. (Accessed 03/23/2013)

3. LODStats. Available at: http://stats.lod2.eu/. (Accessed 25/07/2013)

4. Bizer C, Heath T, Berners-Lee T. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)* 2009, 5:1-22.

5. Shekarpour S, Ngonga Ngomo A-C, Auer Sr. Question answering on interlinked data. In: *Proceedings of the 22nd international conference on World Wide Web*: International World Wide Web Conferences Steering Committee; 2013.

6. Lopez V, Nikolov A, Sabou M, Uren V, Motta E, d'Aquin M. Scaling up question-answering to linked data. In: *Knowledge Engineering and Management by the Masses*: Springer; 2010, 193-210.

7. DBpedia. Available at: http://dbpedia.org/. (Accessed 25/07/2013)

8. GovTrack. Available at: http://www.govtrack.us/. (Accessed 25/07/2013)

9. GeoNames. Available at: http://geonames.org/. (Accessed 25/07/2013)

10. U.S. Census. Available at: http://www.rdfabout.com/demo/census/. (Accessed 25/07/2013)

11. Gruber TR. A translation approach to portable ontology specifications. *Knowledge acquisition* 1993, 5:199-220.

12. Borst WN. *Construction of engineering ontologies for knowledge sharing and reuse*: Universiteit Twente; 1997.

13. Noy NF, McGuinness DL. Ontology development 101: A guide to creating your first ontology. Available at: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html. (Accessed 03/23/2013)

14. Gruber T. What is an Ontology. *Encyclopedia of Database Systems* 2008, 1.

15. Ngomo ACN, Auer S. LIMES: a time-efficient approach for large-scale link discovery on the web of data. In: *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Three*: AAAI Press; 2011.

16. Volz J, Bizer C, Gaedke M, Kobilarov G. Silk–a link discovery framework for the web of data. In: *Proceedings of the 2nd Linked Data on the Web Workshop*: Citeseer; 2009.

17. Araujo S, Hidders J, Schwabe D, De Vries AP. SERIMI-resource description similarity, RDF instance matching and interlinking. *CoRR abs/1107.1104* 2011.

18. Jain P, Hitzler P, Sheth A, Verma K, Yeh P. Ontology alignment for linked open data. *The Semantic Web–ISWC* 2010:402-417.

19. Jain P, Yeh P, Verma K, Vasquez R, Damova M, Hitzler P, Sheth A. Contextual ontology alignment of lod with an upper ontology: A case study with proton. *The Semantic Web: Research and Applications* 2011:80-92.

20. Parundekar R, Knoblock CA, Ambite JL. Discovering concept coverings in ontologies of linked data sources. In: *The Semantic Web - ISWC 2012*: Springer, 427-443.

21. Correndo G, Penta A, Gibbins N, Shadbolt N. Statistical analysis of the owl: sameAs network for aligning concepts in the linking open data cloud. In: *Database and Expert Systems Applications*: Springer; 2012.

22. Gunaratna K, Thirunarayan K, Jain P, Sheth A, Wijeratne S. A statistical and schema independent approach to identify equivalent properties on linked data. In: *Proceedings of the 9th International Conference on Semantic Systems*. Graz, Austria: ACM; 2013.

23. Music Ontology. Available at: http://musicontology.com/. (Accessed 14/10/2013)

24. DBTune MusicBrainz data server. Available at: http://dbtune.org/musicbrainz/. (Accessed 14/10/2013)

25. Miller GA. WordNet: a lexical database for English. *Communications of the ACM* 1995, 38:39-41.

26. Wikipedia. Available at: http://www.wikipedia.org/. (Accessed 25/07/2013)

27. Euzenat Jrm, Shvaiko P. *Ontology matching*: Springer; 2007.

28. Shvaiko P, Euzenat Jrm. Ontology matching: state of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering* 2011.

29. Bellahsene Z, Bonifati A, Rahm E. Schema Matching and Mapping. *Schema Matching and Mapping:, Data-Centric Systems and Applications, ISBN 978-3-642-16517-7. Springer-Verlag Berlin Heidelberg, 2011*, 1.

30. Ontology Alignment Evaluation Initiative. Available at: http://oaei.ontologymatching.org/. (Accessed 25/07/2013)

31. Gruetze T, Bohm C, Naumann F. Holistic and Scalable Ontology Alignment for Linked Open Data. In: *LDOW*; 2012.

32. David J, Guillet F, Briand H. Matching directories and OWL ontologies with AROMA. In: *Conference on Information and Knowledge Management: Proceedings of the 15 th ACM international conference on Information and knowledge management*; 2006.

33. Li J, Tang J, Li Y, Luo Q. RiMOM: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on* 2009, 21:1218-1232.

34. Giunchiglia F, Shvaiko P, Yatskevich M. S-Match: an algorithm and an implementation of semantic matching. *The semantic web: research and applications* 2004:61-75.

35. Freebase. Available at: http://www.freebase.com/. (Accessed 25/07/2013)

36. Terziev I, Kiryakov A, Manov D. D. 1.8. 1 Base upper-level ontology (BULO) Guidance. *Deliverable of EU-IST Project IST* 2005.

37. Cruz IF, Antonelli FP, Stroe C. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment* 2009, 2:1586-1589.

38. Cruz I, Palmonari M, Caimi F, Stroe C. Towards on the go matching of linked open data ontologies. In: *Workshop on Discovering Meaning On The Go in Large Heterogeneous Data*; 2011.

39. Parundekar R, Knoblock C, Ambite J. Linking and building ontologies of linked data. *The Semantic Web - ISWC 2010*:598-614.

40. Nikolov A, Motta E. Capturing emerging relations between schema ontologies on the web of data. 2010.

41. Halpin H, Hayes P, McCusker J, Mcguinness D, Thompson H. When owl: sameas isn't the same: An analysis of identity in linked data. *The Semantic Web–ISWC* 2010:305-320.

42. Suchanek FM, Abiteboul S, Senellart P. PARIS: probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.* 2011, 5:157-168.

43. Tran QV, Ichise R, Ho BQ. Cluster-based similarity aggregation for ontology matching. In: *Proc. of 6th Ontology Matching Workshop*; 2011.

44. Sleeman J, Alonso R, Li H, Pope A, Badia A. Opaque Attribute Alignment. In: *Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on*: IEEE; 2012.

45. Zhao L, Ichise R. Graph-based ontology analysis in the linked open data. In: *Proceedings of the 8th International Conference on Semantic Systems*: ACM; 2012.

46. Franz T, Schultz A, Sizov S, Staab S. Triplerank: Ranking semantic web data by tensor decomposition. *The Semantic Web-ISWC* 2009:213-228.

47. Gunaratna K, Thirunarayan K, Sheth A. Types of Property Pairs and Alignment on Linked Datasets -- A Preliminary Analysis. *Proceedings of the I-SEMANTICS 2013 Posters & Demonstrations Track*:35.

48. Zhang Z, Gentile AL, Blomqvist E, Augenstein I, Ciravegna F. Statistical Knowledge Patterns: Identifying Synonymous Relations in Large Linked Datasets. In: *The Semantic Web-ISWC 2013*: Springer, 703-719.

49. Halpin H, Hayes PJ. When owl: sameAs isn't the same: An analysis of identity links on the semantic web. In: *Linked Data on the Web WWW2010 Workshop (LDOW2010)*; 2010.

50. Volz J, Bizer C, Gaedke M, Kobilarov G. Discovering and maintaining links on the web of data. *The Semantic Web-ISWC* 2009:650-665.

51. Isele R, Jentzsch A, Bizer C. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In: *WebDB*; 2011.

52. Song D, Heflin J. Automatically generating data linkages using a domain-independent candidate selection approach. *The Semantic Web-ISWC 2011*:649-664.

53. Ngonga Ngomo A-C, Lyko K. EAGLE: efficient active learning of link specifications using genetic programming. *The Semantic Web: Research and Applications* 2012:149-163.

54. Ngomo A-CN, Lyko K, Christen V. COALA - Correlation-Aware Active Learning of Link Specifications. In: *The Semantic Web: Semantics and Big Data*: Springer, 442-456.

55. Ngomo A-CN. Link discovery with guaranteed reduction ratio in affine spaces with minkowski measures. In: *The Semantic Web-ISWC 2012*: Springer, 378-393.

56. Isele R, Bizer C. Learning expressive linkage rules using genetic programming. *Proc. VLDB Endow.* 2012, 5:1638-1649.

57. Isele R, Bizer C. Active learning of expressive linkage rules using genetic programming. *Web Semantics: Science, Services and Agents on the World Wide Web* 2013.

58. Joshi AK, Jain P, Hitzler P, Yeh PZ, Verma K, Sheth AP, Damova M. Alignment-based Querying of Linked Open Data. In: *On the Move to Meaningful Internet Systems: OTM 2012*: Springer, 807-824.

59. Jain P, Hitzler P, Verma K, Yeh PZ, Sheth AP. Moving beyond sameAs with PLATO: Partonomy detection for Linked Data. In: *Proceedings of the 23rd ACM conference on Hypertext and social media*: ACM; 2012.

60. Sheth A, Arpinar I, Kashyap V. Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships. *Enhancing the Power of the Internet* 2004:63-94.

61. Prud'Hommeaux E, Seaborne A. SPARQL query language for RDF. Available at: http://www.w3.org/TR/rdf-sparql-query/. (Accessed 23/03/2013)

62. Hartig O, Bizer C, Freytag JC. Executing SPARQL queries over the web of linked data. *The Semantic Web-ISWC* 2009:293-309.

63. Ladwig Gn, Tran T. Linked data query processing strategies. In: *Proceedings of the 9th international semantic web conference on The semantic web-Volume Part I*: Springer-Verlag; 2010.

64. The Linking Open Data cloud diagram. Available at: http://lod-cloud.net/. (Accessed 25/07/2013)

65. Datahub. Available at: http://datahub.io/group/lodcloud. (Accessed 25/07/2013)

66. Harth A, Hose K, Karnstedt M, Polleres A, Sattler KU, Umbrich J. Data summaries for on-demand queries over linked data. In: *Proceedings of the 19th international conference on World wide web*: ACM; 2010.

67. Hose K, Karnstedt M, Koch A, Sattler KU, Zinn D. Processing rank-aware queries in P2P systems. *Databases, Information Systems, and Peer-to-Peer Computing* 2007:171-178.

68. Görlitz O, Staab S. SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In: *Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany*; 2011.

69. Alexander K, Hausenblas M. Describing linked datasets-on the design and usage of void, the'vocabulary of interlinked datasets. In: *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09)*: Citeseer; 2009.

70. Schwarte A, Haase P, Hose K, Schenkel R, Schmidt M. FedX: Optimization techniques for federated query processing on linked data. *The Semantic Web–ISWC 2011* 2011:601-616.

71. Schwarte A, Haase P, Hose K, Schenkel R, Schmidt M. FedX: a federation layer for distributed query processing on linked open data. *The Semanic Web: Research and Applications* 2011:481-486.

72. Konrath M, Gottron T, Scherp A. SchemEX—Web-Scale Indexed Schema Extraction of Linked Open Data. *Semantic Web Challenge, Submission to the Billion Triple Track* 2011.

73. de Oliveira HR, Tavares AT, Lóscio BF. Feedback-based data set recommendation for building linked data applications. In: *Proceedings of the 8th International Conference on Semantic Systems*: ACM; 2012.

74. Nikolov A, d'Aquin M, Motta E. What should I link to? Identifying relevant sources and classes for data linking. *The Semantic Web* 2012:284-299.

75. Tummarello G, Cyganiak R, Catasta M, Danielczyk S, Delbru R, Decker S. Sig. ma: Live views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 2010, 8:355-364.

76. Umbrich Jr, Karnstedt M, Hogan A, Parreira JX. Freshening up while staying fast: Towards hybrid SPARQL queries. In: *Knowledge Engineering and Knowledge Management*: Springer; 2012, 164-174.

77. MusicBrainz. Available at: http://musicbrainz.org/. (Accessed 14/10/2013)

78. ClimbData. Available at: http://datahub.io/dataset/data-incubator-climb. (Accessed 14/10/2013)

79. Lingvoj. Available at: http://www.lingvoj.org/. (Accessed 14/10/2013)

80. Frosterus M, Hyvonen E, Laitio J. Datafinland--a semantic portal for open and linked datasets. In: *The Semanic Web: Research and Applications*: Springer; 2011, 243-254.

81. Lalithsena S, Jain P, Hitzler P, Sheth A. Automatic Domain Identification for Linked Open Data. In: *Proceedings of the 2013 IEEE/WIC/ACM International Conferences on Web Intelligence*. Atlanta, USA; 2013.